Az Országos Commodore Egyesület lapja

C újság

VIII. évfolyam 1993/2



VÉGE a titoknak!

Lapunkban megtalálják pályázatunk eredményhridetését és a győztes programok bemutatását. Bővebb terjedelemben folyamatosan közöljük a díjnyertes munkákat.

MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Pötyögőszolgálatunk valamint a szervizkedvezmény és az apróhirdetés lehetősége tagjaink rendelkezésére áll.

A DEÁKPÁHOLY tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy egy évre 890 Ft.

A PLUSZPÁHOLY tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 2050 Ft.

A SZUPERPÁHOLY tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15x3 db vásárlási utalványt is. Az éves tagsági díj 24 000 Ft.

ÜGYFÉLFOGADÁS: Minden kedden és csütörtökön 12—16 óra között várjuk tagjainkat és az érdeklődőket.

PÖTYÖGŐSZOLGÁLAT: Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvéttel. Postacím: 1388 Budapest 62., Postafiók: 86.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 100 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A C-ÚJSÁG RÉGEBBI SZÁMAI megvásárolhatók az egyesület irodájában, vagy megrendelhetők utánvéttel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 1-76-22-57-es telefonszámon vagy levélben!

Vidéki pluszpáholy-tagjaink háromhavi tikett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C Áruház csomagküldő szolgálatát.

VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:
Baja, AXIS Kft.,
Győri Bartók Béla Művelődési Ház,
Jászberényi Városi Könyvtár,
Kecskemét, SZIGMA—BIT,
Pécsi Apáczai Csere János Gimnázium,
Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:
1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57
Felelős kiadó: Horváth Judit, az egyesület elnöke
Főszerkesztő: Rados Péter, az OCE főtitkára
Felelős szerkesztő: dr. Horváth András
Művészeti szerkesztő: Bausz Sándor
Levélcím: Commodore Újság, 1388 Budapest, 62. Pf.: 86.
Index: ISSN 0237-756 X
Terjeszti a Magyar Posta
Megvásárolható a hírlapárusoknál
92.0305 MSZH Nyomda és Kiadó Kft., Budapest
Felelős vezető: Nagy László

Tagdíjbefizetés

Kedves Tagtársaink!

A realitásokat figyelembe véve egyesületünk közzéteszi, hogy mértéktartóan bár, de megváltoznak tagdíjaink, 1993. január elsejétől.

Deákpáholy:	890 Ft
Pluszpáholy:	2 050 Ft
Szuperpáholy:	24 000 Ft

Országos Commodore Egyesület

Klubdélelőtt

Az Országos Commodore Egyesület klubdélelőttöt tart a Havanna Közösségi Házban minden hónap 3. vasárnapján, délelőtt 9 órától. (1181 Budapest, Kondor Béla sétány 8.)
Megközelíthető: a metró kőbánya–kispesti végállomásától a piros 136-os busszal.
Várunk benneteket! Legközelebb:

február 16-án és március 15-én.

FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!
Kérjük, hogy ha rendelkeznek olyan működő számítógéppel, vagy jó állapotban lévő bármilyen számítógép-tartozékkal, amire nincs szükségük, akkor hozzák el egyesületi irodánkba (minden héten kedden és csütörtökön 11-15 óra között).
Mi összegyűjtjük és eljuttatjuk azokat határainkon túli (erdélyi, kárpátaljai és szlovákiai) magyar iskolákhoz.
Köszönettel:

Országos Commodore Egyesület

Tisztelt Commodore Újság!

Gondolom, sokan próbálkoznak azzal, hogy egy program listáját lekódolják. Én erre tudok egy többé-kevésbé használható eljárást. Ezt a következő programban szemléltetem:

```
1 rem kódoljuk shift és l-betűt!
10 print „ide jön az Ön programja”!!!
```

Ha listázzuk, akkor csak az első sor fog kijönni.

Basa Lionel

Tippek-trükkök nyomtatóhoz

Nyomtató a User-Port-on. 1. tipp

Ha a felhasználói kapura csatlakoztatott nyomtató nem működik, először mindig a kábel bekötését kell megvizsgálni. A nyomtató adatvezetékei (Centronics 2—9) az User-Port C—L érintkezőivel legyenek összekötve. A nyomtató Strobebemenete (Centronics 1) a User-Port M-re jön, míg a nyomtató foglaltságot jelző (BUSY-FLAG) vezetéke (Centronics 10) a User-Port B-re. A nyomtató földelő vezetéke (GND, Centronics 19—30) a C-64 földelő vezetékére (User-Port 1, 12, A vagy N érintkezőire) legyen kötve.

Nyomtató a User-Port-on. 2. tipp

Gyakran előfordul a párhuzamos floppy-adatátviteles rendszereknél, hogy a lemezegység csak akkor működik, ha ugyanakkor a nyomtató összekötő kábele ki van húzva, és fordítva is igaz. Ennek az oka, hogy a C-64 2. jelzővezetékén (User-Port B) a jelek összeütköznek. Itt a segítség viszonylag egyszerű. A floppy, illetőleg a nyomtató eme vezetékeit ne közvetlenül kössük az User-Port csatlakozókra, hanem iktassunk közbe egy-egy diódát (pl. 1N4148). A diódák polaritása feltétlenül figyelembe veendő, az anódjuk nézzen a User-Port felé, míg a katódjuk, amin a jelzőgyűrű van, az nézzen a floppy, illetve a nyomtató felé. Ily módon a diódák a 2. jelző bemenetet alacsony szintre húzzák, egymást azonban kölcsönösen nem zavarják.

Nyomtató a User-port-on. 3. tipp

Ha valaki körszelvényű kábelből fabrikál magának csatlakozást a User-Port és a floppy összekötésére, gyakran csodálkozik, hogy a floppy nem működik, pedig már százszor ellenőrizte. Ennek az oka gyakran a kábel kapacitása. Egy kb. 1 nF kapacitású kondenzátort forrasszunk a User-Port A és B csatlakozói közé. Ez csaknem mindig segít.

Nyomtató a User-Port-on. 4. tip

Ha a nyomtatónak a számítógép hardver-resetjét is végre kell hajtani, az „Init Printer" vezetéket (Centronics 31) egy dióda közbeiktatásával össze kell kötni a User-Porttal. A dióda az anódjával nézzen a nyomtató felé, és megakadályozza, hogy a nyomtató kikapcsolásakor a C64 resetet hajtson végre.

Nyomtató a User-Port-on. 5. tipp

Még egy tipp azoknak, akik a nyomtatót C64-hez is és PC-hez is használják. C64 esetén sok nyomtatóprogramnál az automatikus soremelés DIP-kapcsolójának BE (ON) állásban kell lenni, míg PC-re csatlakoztatva mindig KI (OFF) állás kell. Az újabb nyomtatók többsége ennek az információnak az átvitelére a Centronics 14. érintkezőjét használja. Ha tehát ezeknél a User-Port kábel Centronics dugaszolójának 14. érintkezőjét egy huzaldarabbal áthidaljuk a földelésre (Centronics 19—30), akkor a nyomtatóban az automatikus soremelés DIP-kapcsolója mindig KI (OFF) helyzetben maradhat. Amikor a C64 kábeljét dugjuk a nyomtatóba, az automatikus soremelés be lesz kapcsolva. Vigyázat! Ez a trükk nem minden nyomtatónál eredményes.

Nyomtatás levelezőlapra

Majdnem minden mátrixnyomtató használója nosztalgiával gondol a jó öreg írógépekre, és olykor kénytelen megállapítani, hogy bizonyos előnyeik azoknak is voltak ezekkel szemben. Ezek az előnyök akkor mutatkoznak meg, amikor valamilyen űrlapot kell kitölteni, vagy ha borítékra, levelezőlapra, vékonyabb kartonra kell írni. Azzal a ténnyel, hogy többnyire nem lehet mindig előre látni, hova is kerül a következő betű, a legtöbb felhasználó így vagy úgy kiegyezik.

Amiatt azonban szerfelett mérges lehet bárki, hogy a boríték megcímzése, vagy egy adatlap kitöltése alig lehetséges. Vannak ugyan leporellóra nyomtatott űrlapok, ezek azonban először is drágák, másrészt 1000 példány alatti mennyiség aligha kapható belőlük.

A szokásos példányok méreteik, és többnyire 250 g/m² feletti papírvastagságuk miatt rendszerint fennakadnak a lapadagolón, vagy a henger nem továbbítja őket, vagy ha mégis, akkor ferdén állnak be. Ha mégis kivételesen jól sikerül a behúzás, akkor meg nem lehet pl. a lap alsó soraira írni, mert ott már nem tartja a lapot a papírszorító.

Egyes példányok esetére a következő eljárást ajánlhatjuk. A kartont, borítékot ragasszuk fotosarokkal egy ív lehetőleg vékony papírra vagy leporellóra. A legtöbb nyomtatónál még szükséges lehet a karton felső szegélyét keskeny ragasztószalagcsíkkal is rögzíteni, nehogy a nyomtatóban valahol elakadjon.

Meg kell azonban jegyezni, hogy ez a módszer a nyomtató mechanikájától függően kísérelhető meg. A finomabbak nem biztos, hogy kibírják sérülés nélkül. A robusztusabb nyomtatók (amelyek megengedik 2—3 példány papír + indigó befűzését) esetében azonban a próbálkozás megoldásához is vezethet, mert néhány űrlap kedvéért ezer leporelló-űrlapot készíttetni igencsak költséges mulatság lenne.

„Fontmaster" (betűszerkesztő) LX-800-zal

A „Fontmaster 128" az Epson LX-800-zal és User-Port kábellel az alábbi beállítási adatokkal hajtható meg: Nem kell csatoló (interfész), készülékszám 4, másodlagos című, csatlakozás soros, nyomtató LX-80, opció illesztve, modul 0.

PÁLYÁZAT

Eredményhirdetés

A játékpályázat tanulságai

Stratégiai játékpályázatunk ismét megmozgatta tagságunk és olvasóink fantáziáját. Úgy látszik, nem volt mindenki számára érthető, mit értünk stratégiai játékon (a program valósít meg a játékossal szemben egy stratégiát), mert sok olyan programot kaptunk, ahol a gép két játékos egymás elleni lépéseit adminisztrálja vagy a játékosnak egyedül kell valamilyen feladatot megoldania.

Zsűrink végül 30 pályaművet rangsorolt. A minősítés a pályázati kiírásban szereplő feltételek vizsgálatával, pontozással történt. Így bizony előfordult, hogy egy jól működő, ügyes algoritmusú játék egy gyengébb mögé került, mert szerzője elmulasztott leírást mellékelni programjához. A minősítés szempontjairól, a stratégiai játékpályázat tapasztalatairól külön cikkben szólunk.

Az Országos Commodore Egyesület 1992. évi játékpályázatának

első helyezettje: KOSZPER VILMOS

második helyezettje: BADENSZKY CSABA

A harmadik–tizedik helyezettek:

3. Sasvári Gábor
4. Gazsó János
5. Alexay Péter
6. Szaszkó Péter
7. Szakály György
8. Décsi Tamás
9. Fapál Ferenc
10. Harna M. András

A nyertes programok közlését 1993. februári számunkban kezdjük meg. A későbbiekben a többiek programjait is közzétesszük, sőt a pályázatra beküldött, nem stratégiai játékok is sorra kerülnek.

Szeretnénk felhívni azon olvasóink figyelmét, akik a pályázati kiírást nem olvasták, hogy ezeket a programokat nem lehet a kereskedelmi forgalomban kapható Commodore játékprogramokkal összehasonlítani, hiszen nálunk feltétel volt a BASIC nyelvű, kötött terjedelmű programlista.

A pályázatra több, mint hatvan programot kaptunk, ami sikernek számít, és jelzi az olvasók vállalkozó kedvét. Ehhez viszonyítva nagy csalódást okozott, hogy a beérkezett munkák közel kétharmada nem volt értékelhető, mert nem teljesítette a pályázat kiírásában meghirdetett feltételeket. Ezek között körülbelül egyenlő arányban találtunk olyan programokat, amelyekhez minimális leírás sem tartozott, amelyek egy egyszemélyes logikai játékot valósítanak meg, és amelyekben a gép csak a játék segédeszköze két játékos számára.

Nem tudtunk magyarázatot találni az ilyen tömeges figyelmetlenségre. Sajnos így olyanok munkája is kárba veszett – a pályázat szemszögéből –, akik pedig láthatóan sok fáradságot fordítottak a program tetszetős, technikás kidolgozására, mint például Hargitai Vilmos. Elképzelhető, hogy később néhányat a legjobb érvénytelen programok közül lapunkban bemutatunk.

A stratégiai játékok fogalma valami olyasmit takar, amikor a játékosok, egymás lépéseit figyelembe véve, sok lehetséges lépés közül igyekeznek a legmegfelelőbbet választani, végső soron teret, figurát, lépési lehetőséget véve el a másiktól, a győzelemig. Ilyen játék például a sakk, a malom, az amőba, de még a dominó vagy a Do-guti is; igazán rugalmasan bíráltuk el a különböző játékötleteket. De az olyan program, amellyel nem a számítógépi partner ELLEN játszhatunk egy nem pusztán a szerencsén múló játékot, semmiképp sem volt elfogadható. Az ilyen programban a gép nem „játszik", és szó sem lehet a program által alkalmazott stratégiáról. Mindkét kitétel megtalálható volt a pályázat kiírásában.

Még inkább hiába dolgoztak azok, akik írtak ugyan egy esetleg nagyon jó programot, de a mellékelt dokumentáció nem érte el az elfogadható minimum szintjét. Egy hetekig tartó munkát kár volt még néhány órai fáradság megtakarításával tönkretenni. A pályázat megfogalmazásából világosan kiderül, hogy a programhoz részletes leírást kellett csatolni nemcsak a program használatáról és a játék szabályairól, hanem a stratégia főbb elemeiről is. De ha a stratégia ismertetésétől el is tekinthettünk volna, akkor is túl sok olyan program érkezett, amely leírásába a szerző még az udvariasságból elvárható legkevesebb munkát sem fektette. Egy cédulára firkantott néhány zűrzavaros mondat a játékról még nem tekinthető leírásnak. Úgy látszik, hogy Még többet ésszel! című sorozatunkban érdemes lesz kitérni a programok dokumentálásának alapfogásaira.

A legjobb programok szezőinek körültekintő munkáját jellemezheti, hogy az első öt helyezett pályázó által beküldött leírások közül négy jó minősítést kapott. És sajnáljuk Varga Zoltán és Kesik Lajos fáradozását, akik egy gyengécske dokumentációval is bekerülhettek volna a legjobb tíz közé.

Még egy furcsa dologról essen szó. Több olyan lemezt is kaptunk, amelyről csak keserves, aprólékos munkával tudtuk beolvasni a programokat, és volt olyan is, amely tartalma egyáltalán nem volt megmenthető. Van, aki már alig használható lemezét küldi el hozzánk egy „utolsó útra", ezért ha egy ilyen lemez befuccsol, akkor az nem meglepetés. De van, aki talán újonnan vásárolt, drága lemezt „áldoz" fel, hiába. Feltétlenül el kell oszlatnunk egy félreértést: a High Density (HD) jelzésű lemez nem egyszerűen jobb felbontású (és drágább) a DD (dupla sűrűségű) lemeznél, hanem más fajtájú is. A HD típusú lemez a Commodore lemezegységekben NEM használható, hanem a PC-k 1,2 MB kapacitású lemezegységeihez való! Az ilyen lemezre íráskor többnyire nem kapunk hibajelzést, a felvett

anyag még sincs biztonságban. Egyáltalán nagyon ajánlhatjuk a fontos, vagy beküldött másolatok ellenőrzését visszaolvasással. Meg kell jegyeznünk, hogy a két beolvashatatlan program egyéb okból sem vehetett volna részt az értékelésben.

A játékok témaválasztása nem volt igazán ötletes. Kiugró többségben kaptunk otelló és amőba programokat, de érdekes módon a legjobb pályázatok eredetibb ötletekre alapultak.

Az értékelés során különböző súllyal, de számított az ötletesség, a program megjelenése (kép, hang), a kezelhetőség, a hibavédelem, a szolgáltatások, a jó dokumentáció és a program áttekinthető szerkezete.

Volt, aki megoldotta, hogy a program joystickkel is kezelhető legyen, de ez néha a kezelhetőség kárára vált, a lassúsága miatt. Biztosan sokan mondtak már le egy mutatós játék tartós használatáról a nehézkessége miatt, ezért érdemes egy program kezelését alaposan megtervezni, és főleg sokat próbálgatni. Így kiderülhet az is, milyen kényelmetlen, ha egy játszmát nem lehet félbeszakítani, feladni, amire is csak egy-két programban találtunk lehetőséget.

A hibavédelem leggyengébb pontja a Stop billentyű kikapcsolása, alig néhány program szerzője gondolt erre. Az, hogy a program kezelője ne követhessen el hibát, szabálytalan műveletet, elemi fontosságú, nemcsak a játékprogramokban. Érvényes ez a futás során a lemez használata közben esetleg felmerülő rendszerhibákra is, amelyek kivédésére egy kísérlettel sem találkoztunk.

Szívesen láttunk volna több alapvető, vagy éppen ötletes extra szolgáltatást. Nagyon vonzó lehet például egy feladat teljesítésének számszerű értékelése, és a legjobb eredmények kimenthetősége lemezre vagy kazettára. Nem sokan próbálkoztak meg a magyar szöveg ékezetes megjelenítésével, és csak alig néhányan új karakterkészlet beépítésével, pedig ennek módszerei már régen ki vannak dogozva.

A pályázat egyik fő célja volt áttekinthető és a többi olvasó által jól érthető szerkezetű programok megjelenítése. Így tehát jelentős előnyt élvezett a REM utasításokkal tördelt, kevés hosszú ugrással tarkított, ésszerű felépítésű program. Kellemes meglepetésünkre a legjobb helyezéseket elért programok között találhatók mintaszerűen megszerkesztettek is, mindenki okulására (Koszper Vilmos, Alexay Péter, Szaszkó Péter munkái). Sajnos szinte mindenki ránk hagyta a programok szövegének szóközökkel való tagolását, ami megjelenéskor pedig előnyös. Bár nézetünk szerint felesleges a BASIC szöveget kettőspontokkal, vagy más módon beljebb kezdett sorokkal tagolni, mégis dicséretet érdemel, aki ilyen munkára sem sajnálta az erejét.

Sajnos megfigyelhetjük, hogy a kiírásnak az a része, amely megtiltotta gépi kódú programok beillesztését, sajátos beállítottságot eredményezett a pályázókban. Volt, aki ki is jelentette, hogy még POKE utasítást sem akart emiatt alkalmazni, ennek következtében elmulasztva a háttérszín beállítását is. A gépi kódú program és a BASIC-ben közvetlenül nem kezelhetően elhelyezett adatok két teljesen eltérő fogalomcsoport. Egyáltalán nem sértette volna a pályázat szabályait az a nagyon praktikus megoldás, hogy a program a karakterkészlet vagy sprite-ok adatait nem generálná hosszasan a DATA sorokból, hanem készen töltené be lemezről. Ezzel a technikával akár nagyfelbontású grafikus háttereket is lehetett volna használni, mivel ezek előkészítése nem volt BASIC-ben kötelező. Egy sprite adatait nem nevezhetjük gépi kódú programnak.

Itt vissza kell térnünk egy pillanatra a pályázat feltételeinek betartásához. A programok elbírálói között heves vitára adott okot három dolgozat egy-egy kis hiányossága. Badenszky Csaba Táblás játéka tartalmaz egy SYS utasítást, amely a kurzor helyének beállítására szolgál. Mivel a program egyéb tekintetben határozottan jól sikerült, nem tűnt teljesen igazságosnak a dolgozat kizárása, főleg mert a hívott szubrutin egyenértékűen kiváltható egy-két BASIC sorral. Büntetést érdemel a figyelmetlenségéért, így a megismételt összevetésben a program már súlyos mínuszokkal vett részt, ennek ellenére 2. helyen végzett. Hasonlóan hátránnyal indult Szaszkó Péter Zsírozása, amelyhez a szerző gondosan kidolgozott dokumentációt mellékelt, az algoritmus bemutatása nélkül; és Décsi Tamás Reversije, amely az algoritmust szintén nem tárgyalja meggyőzően. A háborgó kedélyek lecsillapítására leszögezzük, hogy ilyen engedményt csak kiemelkedően jó pályázatok esetében tartottunk méltányosnak.

Összességében megállapítható, hogy a pályázat elérte a célját, és a legjobb helyezéseket azokra valóban méltó programok között oszthattuk el. Kár, hogy nem okozott nagyobb gondot a legjobbak kiválasztása. Ennek ellenére olvasóink néhány valóban érdekes, tanulságos és új játékprogrammal lehetnek gazdagabbak, és remélhetőleg sok hasznos tapasztalattal a BASIC nyelv professzionális programozásáról.

AZ ELSŐ

Szomszéd játék

Programnyelv: *C64 BASIC és C PLUS/4 BASIC* közös része.

Gép: *Fut C64-en és C PLUS/4-en.*

A játékról Vargha Balázs könyveiben lehet olvasni (pl. Játékkoktél). A játék megvalósított változata kissé eltér a könyvben leírttól.

Szabályai:

A játékot egy nyolcszor nyolc négyzetből álló táblán játssza két játékos.

A tábla kezdetben üres. A játékosok felváltva lépnek. A lépés abból áll, hogy a játékos a táblán egy négyzetlapkával letakar egy mezőt. Ez a lapocska aztán a játék végéig a helyén marad. A játékos nem teheti tetszés szerinti helyre a lapocskát. Először is nem teheti olyan mezőre, ahol már van lapocska. És nem teheti olyan mezőre sem (a képernyő és a karakterek színét nem állítja be), amelynek közvetlen, oldalával érintkező *szomszédja* már le van takarva.

```
10 REM SZOMSZED JATEK
20 REM KOSZPER VILMOS,1992
30 REM 1992.7.19-1992.9.30
40 :
180 :
190 REM********************************
200 REM FO ELOKESZITO RESZ
210 GOSUB470
220 GOSUB760
230 GOSUB860
240 GOSUB1120
250 REM********************************
260 REM********************************
270 REM FORESZ
280 :IF PX>0 THEN350
290 GOSUB2340
300 GOSUB8010
310 GOSUB8080
320 GOTO280
330 REM********************************
340 REM********************************
350 :REM BEFEJEZES
360 PRINT "";
370 X=5:Y=10
380 TX$=" ,  , ,             , ,"
390 GOSUB1160
400 X=5:Y=11
410 TX$="JO JATEK VOLT! VISZONTLATASRA!"
420 GOSUB1160
430 REM********************************
440 :
450 END
460 :
470 :REM DEKLARACIOK
480 PLUS4=0:REM PLUS4 VAGY C64
490 P=0:REM VALOSZINUSEG
500 AB=0:REM EPPEN MELYIK JATEKOS
510 SZ=0:WE=0:LP=0:REM SZAM,ERTEK,LEPTETO
520 X=0:Y=0:XT=0:YT=0:XC=0:YC=0:XI=0:YI=0:XQ=0:YQ=0:REM KEP- ES TABLAKOORDINATAK
530 JX=0:SX=0:PX=0:REM FELTETELEK
540 Q=0:QQ=0:Q$="":J=0:REM ATMENETI
550 KL=0:REM KOTELEZO LEPES
560 F1=0:F2=0:F3=0:F4=0:D1=0:D2=0:D3=0:D4=0:BS=0:REM ERTEKELO FV.-HEZ
570 DIM TA(9,9),TB(8,8):REM A TABLA BELSO MEGJELENITESE
580 DIM JK(1),JL(1),SK(1),SL(1),PM(1),PP(1),NY(1)
590 DIM PE(1),ME(4),MO(4),MB(4),LW(2,2)
600 DIM JA$(6):REM SZOVEGKONST
610 DIM GY(11,1):REM GYURUKHOZ
620 FOR J=0 TO 11:READ GY(J,0),GY(J,1):NEXT J
630 JA$(0)="EMBER"
640 JA$(1)="P-PRG"
650 JA$(2)="Q-PRG"
660 JA$(3)="R-PRG"
670 JA$(4)="S1PRG"
680 JA$(5)="S2PRG"
690 JA$(6)="S3PRG"
700 JL$(0)="ODA---------"
710 JL$(1)="------VISSZA"
720 JL$(2)="------------"
730 RETURN
740 DATA -1,0,0,-1,0,1,1,0,-1,-1,-1,1,1,-1,1,1,-2,0,0,-2,0,2,2,0
750 :
760 :REM PLUS4/C64
770 PRINT CHR$(142);CHR$(8);
780 PRINT "A";
790 Q=PEEK(3072)
800 PRINT "B";
810 XC=211
820 IF Q=1 AND PEEK(3072)=2 THEN PLUS4=1:XC=202
830 YC=XC+3
840 RETURN
850 :
860 :REM ALLANDOABRA
870 PRINT "";
880 PRINT"               ,  , ,"
890 PRINT"   SZOMSZED  JATEK     A      B     "
900 PRINT"                     ,"
910 PRINT"  ┌┬┬┬┬┬┬┬┬┬┬┐   JATSZ:"
920 PRINT"  │ │ │ │ │ │ │ │ │ │  , ,"
```

A játék akkor fejeződik be, ha már a játékosok nem tudnak új lapocskát letenni. A játék végén a táblán szabad és lefedett négyzetek láthatók rendszertelen eloszlásban. Mivel a két játékos egyforma lapocskát használt, azt egyáltalán nem tudjuk, hogy menet közben ki melyiket rakta le, de ez nem is érdekes. Az üres mezőket kell nézni. Egy-egy üres mező letakart *szomszédainak* száma egy, kettő, három vagy négy.

Az egyik játékos törekedhet arra, hogy például minél több olyan üres mező legyen, melynek csak egyetlen letakart *szomszédja* van, míg a másik arra, hogy a négy *szomszédos* üres mezők legyenek sokan. Ebben a két játékosnak a játék elkezdése előtt kell megállapodnia, csak az a lényeg, hogy különböző számú szomszédot válasszanak. Az a játékos nyer, amelyiknek a játék végén több üres mezeje lesz, a maga választotta számú szomszéddal. Az esélyek nem egyenlőek, ezért oda- és visszavágót kell játszani.

A program használata

A program kirajzolja az üres játéktáblát. A tábla mellett jobb oldalon a két játékosra (A, B) két oszlopba írja a vonatkozó adatokat.

```
930 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤  ■JATEK:■
940 PRINT"  │ │ │ │ │ │ │ │ │ │"
950 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤  ■SZOM: ■
960 PRINT"  │ │ │ │ │ │ │ │ │ │"
970 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤  ■MOST: ■
980 PRINT"  │ │ │ │ │ │ │ │ │ │"
990 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤  ■PONT: ■
1000 PRINT"  │ │ │ │ │ │ │ │ │ │"
1010 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤  ■NYERT:■
1020 PRINT"  │ │ │ │ │ │ │ │ │ │"
1030 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤"
1040 PRINT"  │ │ │ │ │ │ │ │ │ │"
1050 PRINT"  ├─┼─┼─┼─┼─┼─┼─┼─┼─┤   ";
1060 IF PLUS4>0 THEN PRINT "PLUS/4":GOTO1030
1070 PRINT"C64"
1080 :PRINT"  │ │ │ │ │ │ │ │ │ │"
1090 PRINT"  └─┴─┴─┴─┴─┴─┴─┴─┴─┘"
1100 RETURN
1110 :
1120 :REM EGYEB ELOKESZITES
1130 PX=0:REM PRG KILEPESI FELTETEL
1140 RETURN
1150 :
1160 :REM SZOVEG X,Y,TX$
1170 IF XC=0 OR YC=0 THEN STOP:REM KESOBB KIIKTATANDO ELLENORZES
1180 POKE XC,X:POKE YC,Y:PRINT "■■";TX$;
1190 RETURN
1200 :
1210 :REM UZENET A JATSZOKROL
1220 X=1:Y=21:TX$="A JATSZOK KIVALASZTASA."
1230 GOSUB1160
1240 X=1:Y=20:TX$="    ,    ,      ,     ,"
1250 GOSUB1160
1260 RETURN
1270 :
1280 :REM UZENET TORLESE
1290 X=1:TX$="                                       "
1300 Y=21:GOSUB1160
1310 Y=20:GOSUB1160
1320 RETURN
1330 :
1340 :REM AKTU SZOVEG TORLESE
1350 X=1:TX$=""
1360 Y=23:GOSUB1160
1370 Y=22:GOSUB1160
1380 RETURN
1390 :
1400 :REM JKIRAS
1410 TX$=JA$(JK(AB))
1420 Y=3:X=28+AB*6
1430 GOSUB1160
1440 RETURN
1450 :
1460 :REM * Q SZERINT
1470 TX$=" "
1480 X=1+(1-Q)*12:Y=23
1490 GOSUB1160
1500 TX$="*"
1510 X=1+Q*12:Y=23
1520 GOSUB1160
1530 RETURN
1540 :
1550 :REM PRG QQ SZERINT
1560 TX$=JA$(QQ)
1570 X=14:Y=23
1580 GOSUB1160
1590 RETURN
1600 :
1610 :REM JATEKOS AB SZERINT
1620 TX$="■"+CHR$(65+AB)+"■"
1630 X=2:Y=23
1640 GOSUB1160
1650 RETURN
1660 :
1670 :REM INPJATSZIE
1680 GOSUB1460
1690 GOSUB1550
1700 GOSUB1610
1710 Q$=""
1720 RETURN
```

Minden szükséges adatot a következő billentyűkkel lehet megadni:

kurzorvezető nyilak	(adatmódosítás)
return	(adatok véglegesítése)
X	(továbblépés)
Y	új sorozata kezdődik a játékoknak
W	vége a játéknak.

A jobb oldalon az adatok jelentése:

JÁTSSZ: a játékos kijelölése

lehet: EMBER emberi játékos

P-PRG, Q-PRG, R-PRG, S1PRG, S2PRG, S3PRG – különböző gépi stratégiák

teljesen szabadon lehet választani, mindkét játékos lehet gépi program is vagy ember.

JÁTÉK: jelzi, hogy odavágó-e vagy visszavágó.

SZOM.: itt jelöli, hogy melyik játékos hány *szomszédos* üres négyzetet választott.

MOST: az aktuális játszma pontjainak száma.

PONT: az odavágó és visszavágó összetett pontjainak száma.

NYERT: amíg a játékosok azonosak, gyűjti, hogy ki hány játszmát nyert (győzelem = 2 pont).

A képernyő alsó sorai az üzeneteket mutatják.

* * *

A gép stratégiái

P-PRG. Kezdete: 4240. sor.

XT és YT fogja tartalmazni a ,,kiszámított'' koordinátákat. Teljesen véletlenszerűen választja ki a lehetséges helyek közül azt, ahova le kell rakni a következő lapot.

A 4300. sorban van egy rövidzár. A ME(0) változó tartalmazza azoknak az üres mezőknek a számát, amelyeknek még nincs *szomszédja*, vagyis ahova még le lehet rakni. Ha ez kisebb 5-nél, a legelső lehetséges helyre lerakja a fekete vagy sötét lapkát.

Q-PRG. Kezdete: 4400. sor.

Mindjárt a 4410. sorban vagy egy elugrás a P-PRG-ra, a RANDOM stratégiára. Ez a feltételes elugrás a KL tartalmától függ (KL = kötelező lerakni). A KL akkor válik nem zéróvá, ha a táblán már *csak* olyan szabad mezők vannak, ahova

```
1730 :
1740 :REM Q=0 ELJ
1750 AB=1-AB
1760 GOSUB1610
1770 RETURN
1780 :
1790 :REM QQ-
1800 QQ=QQ-1
1810 IF QQ<0 THEN QQ=6
1820 GOSUB1550
1830 RETURN
1840 :
1850 :REM QQ+
1860 QQ=QQ+1
1870 IF QQ>6 THEN QQ=0
1880 GOSUB1550
1890 RETURN
1900 :
1910 :REM FEL1
1920 IF Q=0 THEN GOSUB1740 :RETURN
1930 GOSUB1790
1940 RETURN
1950 :
1960 :REM LE1
1970 IF Q=0 THEN GOSUB1740 :RETURN
1980 GOSUB1850
1990 RETURN
2000 :
2010 :REM SZETVALOGATAS
2020 IF Q$="□ THEN GOSUB1910 :RETURN
2030 IF Q$="■ THEN GOSUB1960 :RETURN
2040 IF Q$="■ THEN Q=1:GOSUB1460 :RETURN
2050 IF Q$="■ THEN Q=0:GOSUB1460 :RETURN
2060 IF Q$=CHR$(13) THEN JK(AB)=QQ:GOSUB1400
2070 RETURN
2080 :
2090 :REM INPJATSZ1T
2100 :IF Q$="X" OR Q$=CHR$(13) THEN RETURN
2110 GET Q$
2120 GOSUB2010
2130 GOTO2100
2140 :
2150 :REM INPJATSZ1
2160 :IF Q$="X" THEN RETURN
2170 GOSUB1670
2180 GOSUB2090
2190 GOTO2160
2200 :
2210 :REM JATSZOK, UJ SOROZAT
2220 FOR AB=0 TO 1
2230 GOSUB1400
2240 NEXT AB
2250 GOSUB3000
2260 Q=0:QQ=0:Q$="":AB=0
2270 X=4:Y=23:TX$="JATEKOS:          (NYILAK, RETURN, X)"
2280 GOSUB1160
2290 X=4:Y=22:TX$=" , ,"
2300 GOSUB1160
2310 GOSUB2150
2320 RETURN
2330 :
2340 :REM JMASOROZAT ELOKESZITO
2350 SX=0:REM SOROZAT KILEPESI FELTETEL
2360 NY(0)=0:NY(1)=0:SK(0)=1:SK(1)=1
2370 GOSUB1210
2380 GOSUB2210
2390 Q$=""
2400 GOSUB1280
2410 GOSUB1340
2420 GOSUB2450
2430 RETURN
2440 :
2450 :REM NYERT
2460 X=29:Y=13:TX$=STR$(NY(0))
2470 TX$=TX$+"   "
2480 TX$=LEFT$(TX$,4)
2490 GOSUB1160
2500 X=35:Y=13:TX$=STR$(NY(1))
2510 TX$=TX$+"   "
2520 TX$=LEFT$(TX$,4)
```

előbb-utóbb mindenképpen le kell rakni a lapkát. (Ezek azok a mezők, amelyek még üresek, nincs letakart *szomszédjuk*, de mégsem lehet melléjük lerakni, mert üres *szomszédaik* már tiltott mezők.)

A Q-PRG stratégiája nagyon egyszerű. Megnézi, hogyan változik meg a tábla képe (saját szempontjából) a legelőnyösebben, ha a következő lépésben lerak. Megvizsgálja az összes lehetséges lépést és egy háromsoros tárolóba rakja a legjobbakat (LW mátrix). Ha a legjobb lépésekből több van, mint három, akkor csak az első hármat hagyja meg. Ha végigvizsgálta a táblát, véletlenszerűen választ a három legjobb közül.

LW (P, 0) felel meg az X koordinátának
LW (P, 1) — ״ — Y — ״ —
LW (P, 2)-ben van az érték (a lépés értéke)

A Q-PRG szűk látókörű (bár jobb mint a P-PRG), mert csak egy lépést lát előre.

Az *P-PRG* hasonló a Q-PRG-hez. (Kezdete 4900). Egy különbség azért van. Mielőtt végigvizsgálná az összes lépéslehetőséget, a kötelező lépéseket végrehajtja (de természetesen a számítások után visszaveszi).

Az *S1PRG, S2PRG, S3PRG* lényegüket tekintve egyformák, a köztük lévő különbséget a kiértékelés számszerűsége adja. (Kezdet: 5380, 5410, 5440.)

Az S stratégiák — először végrehajtják a kötelező lépéseket (melyeket a számítás végeztével visszavesznek). Azután megvizsgálják az összes lehetséges lépést, de nemcsak a közvetlen *szomszéd*, hanem a *szomszéd* szomszédját is figyelembe véve.

A kiértékelőben (MEZOERTEKELES 5550. sor) a *szomszédok szomszédait* az F1, F2, F3, F4, D1, D2, D3, D4 tekintetbe vételével, a *szomszédokat* a BS tekintetbe vételével veszi figyelembe.

Megjegyzés a programlistához:

Kettőspontok tagolják, elválasztják a modulokat. Kettőspont vezeti be azt a sort, amelyre hivatkozás történik GOTO-val, GOSUB-bal, THEN-nel.

Koszper Vilmos

```
2530 GOSUB1160
2540 RETURN
2550 :
2560 :REM JL$IR
2570 X=28:Y=5:TX$=JL$(JJ)
2580 GOSUB1160
2590 RETURN
2600 :
2610 :REM TORLESEK
2620 FOR AB=0 TO 1
2630 PP(AB)=0:PM(AB)=0
2640 NEXT AB
2650 RETURN
2660 :
2670 :REM SLIR
2680 TX$=CHR$(48+SL(JJ))
2690 X=30+JJ*6:Y=7
2700 GOSUB1160
2710 RETURN
2720 :
2730 :REM PMIR
2740 TX$=STR$(PM(JJ))
2750 GOSUB2870
2760 X=29+JJ*6:Y=9
2770 GOSUB1160
2780 RETURN
2790 :
2800 :REM PPIR
2810 TX$=STR$(PP(JJ))
2820 GOSUB2870
2830 X=29+JJ*6:Y=11
2840 GOSUB1160
2850 RETURN
2860 :
2870 :REM TX ADJUSZT
2880 TX$="00"+TX$
2890 TX$=RIGHT$(TX$,2)
2900 RETURN
2910 :
2920 :REM MINDIR
2930 FOR JJ=0 TO 1
2940 GOSUB2670
2950 GOSUB2730
2960 GOSUB2800
2970 NEXT JJ
2980 RETURN
2990 :
3000 :REM --IR
3010 JJ=2:GOSUB2560
3020 RETURN
3030 :
3040 :REM JATSZMAPAR ELOKESZITO
3050 GOSUB2610
3060 GOSUB3000
3070 GOSUB2920
3080 GOSUB8550
3090 RETURN
3100 :
3110 :REM ODAELOKESZ
3120 JJ=0:SL(0)=SK(0):SL(1)=SK(1)
3130 PE(0)=0:PE(1)=0
3140 GOTO3190
3150 :
3160 :REM VISSZAELOKESZ
3170 JJ=1:SL(0)=SK(1):SL(1)=SK(0)
3180 PE(0)=PM(0):PE(1)=PM(1)
3190 :JX=0:KL=0:REM KOTELEZO LEPES
3200 TX$="NEHANY MASODPERC VARAKOZAS!"
3210 X=1:Y=21:GOSUB1160
3220 TX$="   ,   ,       ,       ,"
3230 Y=20:GOSUB1160
3240 FOR J=1 TO 4
3250 ME(J)=0
3260 NEXT J
3270 ME(0)=64
3280 PM(0)=0:PM(1)=0
3290 FOR J=0 TO 9
3300 TA(J,0)=20:TA(0,J)=20:TA(J,9)=20:TA(9,J)=20
3310 NEXT J
3320 TX$=""
3330 FOR YI=1 TO 8
3340 FOR XI=1 TO 8
3350 TA(XI,YI)=0:GOSUB3660
3360 NEXT XI
3370 NEXT YI
3380 GOSUB2560
3390 GOSUB2920
3400 GOSUB1280
3410 RETURN
3420 :
3430 :
3440 TX$="■+CHR$(65+AB)+"■ JATEKOS ("+JA$(JK(AB))+") LEP."
3450 X=2:Y=23:GOSUB1160
3460 TX$="     ,  ,              ,"
3470 X=2:Y=22:GOSUB1160
3480 RETURN
3490 :
3500 :REM KURZORT IR
3510 TX$="▒"
3520 IF TA(XI,YI)=0 THEN TX$="♦"
3530 GOSUB3660
3540 RETURN
3550 :
3560 :REM LAPKAT IR
3570 TX$="■ ■
3580 GOSUB3660
3590 RETURN
3600 :
3610 :REM SZOKOZT IR
3620 TX$=" "
3630 GOSUB3660
3640 RETURN
3650 :
3660 :REM A TABLARA IR
3670 X=XI*2+1:Y=YI*2+2
3680 GOSUB1160
3690 RETURN
3700 :
3710 :REM ODEBB A KZ
3720 Q=TA(XI,YI)
3730 IF Q=0 THEN GOSUB3610 :RETURN
3740 IF Q=10 THEN GOSUB3560 :RETURN
3750 TX$=CHR$(48+Q)
3760 GOSUB3660
3770 RETURN
3780 :
3790 :REM INPUT KOZEPE
3800 YI=1
3810 :XI=1
3820 :IF TA(XI,YI)>0 THEN4050
3830 :GOSUB3500
3840 :GET Q$
3850 IF Q$="■ THEN3930
3860 IF Q$="■ THEN3960
3870 IF Q$="⊐ THEN3990
3880 IF Q$="■ THEN4020
3890 IF Q$<>CHR$(13) THEN3840
3900 IF TA(XI,YI)>0 THEN3840
3910 XT=XI:YT=YI
3920 RETURN
3930 :IF XI>7 THEN3840
3940 GOSUB3710 :XI=XI+1
3950 GOTO3830
3960 :IF XI<2 THEN3840
3970 GOSUB3710 :XI=XI-1
3980 GOTO3830
3990 :IF YI<2 THEN3840
4000 GOSUB3710 :YI=YI-1
4010 GOTO3830
4020 :IF YI>7 THEN3840
4030 GOSUB3710 :YI=YI+1
4040 GOTO3830
4050 :XI=XI+1
4060 IF XI<=8 THEN3820
```

```
4070 YI=YI+1
4080 IF YI>8 THEN STOP
4090 GOTO3810
4100 :
4110 :REM INPUT ELOK
4120 X=1:Y=21:TX$="NYILAK=HELYKERESES, RETURN=IDE!"
4130 GOSUB1160
4140 X=1:Y=20:TX$="                              ,"
4150 GOSUB1160
4160 RETURN
4170 :
4180 :REM INPUT
4190 GOSUB4110
4200 GOSUB3790
4210 GOSUB1280
4220 RETURN
4230 :
4240 :REM P-PRG (RANDOM .1)
4250 YI=1+INT(8*RND(8))
4260 XI=1+INT(8*RND(8)):GOTO4290
4270 :YI=1
4280 :XI=1
4290 :IF TA(XI,YI)>0 THEN4340
4300 IF ME(0)<5 OR KL>0 THEN4320
4310 P=RND(5):IF P>.1 THEN4340
4320 :XT=XI:YT=YI
4330 RETURN
4340 :XI=XI+1
4350 IF XI<=8 THEN4290
4360 YI=YI+1
4370 IF YI>8 THEN4270
4380 GOTO4280
4390 :
4400 :REM Q-PRG
4410 IF KL>0 THEN4240
4420 GOSUB4620
4430 FOR YI=1 TO 8
4440 FOR XI=1 TO 8
4450 IF TA(XI,YI)>0 THEN4520
4460 SZ=TA(XI+1,YI):GOSUB4680
4470 SZ=TA(XI-1,YI):GOSUB4680
4480 SZ=TA(XI,YI+1):GOSUB4680
4490 SZ=TA(XI,YI-1):GOSUB4680
4500 GOSUB4810
4510 GOSUB4740
4520 :NEXT XI
4530 NEXT YI
4540 GOSUB4570
4550 RETURN
4560 :
4570 :LP=LP+1:IF LP>3 THEN LP=3:REM LKIVALASZT
4580 P=INT(LP*RND(5))
4590 XT=LW(P,0):YT=LW(P,1)
4600 RETURN
4610 :
4620 :REM LWTORL
4630 FOR SZ=0 TO 2
4640 LW(SZ,2)=-1000
4650 NEXT SZ
4660 RETURN
4670 :
4680 :REM MEBE
4690 IF SZ>3 THEN RETURN
4700 ME(SZ+1)=ME(SZ+1)+1
4710 MO(SZ+1)=MO(SZ+1)+1
4720 RETURN
4730 :
4740 :REM MEVISSZA
4750 FOR SZ=1 TO 4
4760 ME(SZ)=ME(SZ)-MO(SZ)
4770 MO(SZ)=0
4780 NEXT SZ
4790 RETURN
4800 :
4810 :REM ERTEKJEGYZES
4820 WE=ME(SL(AB))-ME(SL(1-AB))
4830 :IF WE<LW(0,2) THEN RETURN
4840 LP=LP+1
4850 IF WE>LW(0,2) THEN LP=0
4860 IF LP>2 THEN RETURN
4870 LW(LP,0)=XI:LW(LP,1)=YI:LW(LP,2)=WE
4880 RETURN
4890 :
4900 :REM R-PRG
4910 IF KL>0 THEN4240
4920 GOSUB5090
4930 IF ME(0)<1 THEN GOSUB4980 :GOSUB4240 :RETURN
4940 GOSUB4400
4950 GOSUB4980
4960 RETURN
4970 :
4980 :REM VISSZAALLIT
4990 FOR SZ=0 TO 4
5000 ME(SZ)=MB(SZ)
5010 NEXT SZ
5020 FOR YI=1 TO 8
5030 FOR XI=1 TO 8
5040 TA(XI,YI)=TB(XI,YI)
5050 NEXT XI
5060 NEXT YI
5070 RETURN
5080 :
5090 :REM BEILLESZTES
5100 FOR SZ=0 TO 4
5110 MB(SZ)=ME(SZ)
5120 NEXT SZ
5130 FOR YI=1 TO 8
5140 FOR XI=1 TO 8
5150 TB(XI,YI)=TA(XI,YI)
5160 NEXT XI
5170 NEXT YI
5180 FOR YI=1 TO 8
5190 FOR XI=1 TO 8
5200 IF TA(XI,YI)>0 THEN5260
5210 IF TA(XI-1,YI)=0 THEN5260
5220 IF TA(XI+1,YI)=0 THEN5260
5230 IF TA(XI,YI-1)=0 THEN5260
5240 IF TA(XI,YI+1)=0 THEN5260
5250 GOSUB5300
5260 :NEXT XI
5270 NEXT YI
5280 RETURN
5290 :
5300 :REM LERAKAS
5310 TA(XI,YI)=10:ME(0)=ME(0)-1
5320 XQ=XI-1:YQ=YI:GOSUB6250
5330 XQ=XI+1:YQ=YI:GOSUB6250
5340 XQ=XI:YQ=YI-1:GOSUB6250
5350 XQ=XI:YQ=YI+1:GOSUB6250
5360 RETURN
5370 :
5380 :REM S1PRG
```

```
5390 F1=1:F2=2:F3=3:F4=4:D1=-1:D2=-2:D3=-3:D4=-4:BS=10:REM ERTEKELO FV.-HEZ
5400 :
5410 :REM S2PRG
5420 F1=1:F2=1:F3=1:F4=1:D1=-1:D2=-1:D3=-1:D4=-1:BS=10:REM ERTEKELO FV.-HEZ
5430 :
5440 :REM S3PRG
5450 F1=0:F2=0:F3=0:F4=0:D1=0:D2=0:D3=0:D4=0:BS=1:REM ERTEKELO FV.-HEZ
5460 IF KL>0 THEN4240
5470 GOSUB4620
5480 GOSUB5090
5490 IF ME(0)<1 THEN KL=1:GOSUB4980 :GOSUB4240 :RETURN
5500 GOSUB5830
5510 GOSUB4570
5520 GOSUB4980
5530 RETURN
5540 :
5550 :REM MEZOERTEKELES
5560 EU=0
5570 FOR J=0 TO 11
5580 XQ=GY(J,0)
5590 IF XQ<1 OR XQ>8 THEN5800
5600 YQ=GY(J,1)
5610 IF YQ<1 OR YQ>8 THEN5800
5620 SZ=TA(XQ,YQ)
5630 CS=SGN(TA(XQ+1,YQ))
5640 P=SGN(TA(XQ-1,YQ)):CS=CS+P
5650 P=SGN(TA(XQ,YQ+1)):CS=CS+P
5660 P=SGN(TA(XQ,YQ-1)):CS=CS+P
5670 IF CS=4 THEN5780
5680 IF SZ<SL(AB) THEN5750
5690 IF SZ>SL(AB) THEN5720
5700 IF SZ>SL(1-AB) THEN EU=EU+F3:GOTO5800
5710 EU=EU+D2:GOTO5800
5720 :IF SZ>SL(1-AB) THEN EU=EU+D1:GOTO5800
5730 IF SZ<SL(1-AB) THEN EU=EU+D4:GOTO5800
5740 EU=EU+D3:GOTO5800
5750 :IF SZ<SL(1-AB) THEN EU=EU+F1:GOTO5800
5760 IF SZ>SL(AB) THEN EU=EU+F4:GOTO5800
5770 EU=EU+F2:GOTO5800
5780 :IF SZ=SL(AB) THEN EU=EU+BS:GOTO5800
5790 IF SZ=SL(1-AB) THEN EU=EU-BS
5800 :NEXT J
5810 RETURN
5820 :
5830 :REM KETGYURU
5840 FOR YI=1 TO 8
5850 FOR XI=1 TO 8
5860 IF TA(XI,YI)>0 THEN5940
5870 GOSUB5550
5880 E=EU
5890 GOSUB5300
5900 GOSUB5550
5910 WE=EU-E
5920 GOSUB4830
5930 GOSUB5980
5940 :NEXT XI
5950 NEXT YI
5960 RETURN
5970 :
5980 :REM VISSZAVETEL
5990 TA(XI,YI)=0
6000 XQ=XI:YQ=YI-1:GOSUB6050
6010 XQ=XI:YQ=YI+1:GOSUB6050
6020 XQ=XI+1:YQ=YI:GOSUB6050
6030 XQ=XI-1:YQ=YI:GOSUB6050
6040 RETURN
6050 :IF TA(XQ,YQ)>5 THEN RETURN
6060 TA(XQ,YQ)=TA(XQ,YQ)-1
6070 RETURN
6080 :
6090 :REM LEPES
6100 IF JK(AB)=0 THEN GOSUB4180 :RETURN
6110 IF JK(AB)=1 THEN GOSUB4240 :RETURN
6120 IF JK(AB)=2 THEN GOSUB4400 :RETURN
6130 IF JK(AB)=3 THEN GOSUB4900 :RETURN
6140 IF JK(AB)=4 THEN GOSUB5380 :RETURN
6150 IF JK(AB)=5 THEN GOSUB5410 :RETURN
6160 IF JK(AB)=6 THEN GOSUB5440 :RETURN
6170 RETURN
6180 :
```

```
6190 :REM LEPES ADMINISZTRACIO RESZLETE
6200 GOSUB6250
6210 IF Q>9 THEN RETURN
6220 GOSUB6330
6230 RETURN
6240 :
6250 :Q=TA(XQ,YQ)
6260 IF Q>9 THEN RETURN
6270 ME(Q)=ME(Q)-1
6280 Q=Q+1
6290 TA(XQ,YQ)=Q
6300 ME(Q)=ME(Q)+1
6310 RETURN
6320 :
6330 :X=XQ*2+1:Y=YQ*2+2:TX$=CHR$(48+Q)
6340 GOSUB1160
6350 RETURN
6360 :
6370 :REM LEPES ADMINISZTRACIO
6380 TA(XT,YT)=10
6390 XI=XT:YI=YT:TX$="♦":GOSUB3660
6400 ME(0)=ME(0)-1
6410 XQ=XT:YQ=YT+1
6420 GOSUB6190
6430 XQ=XT:YQ=YT-1
6440 GOSUB6190
6450 XQ=XT-1:YQ=YT
6460 GOSUB6190
6470 XQ=XT+1:YQ=YT
6480 GOSUB6190
6490 FOR J=0 TO 1
6500 PM(J)=ME(SL(J))
6510 PP(J)=PM(J)+PE(J)
6520 NEXT J
6530 XI=XT:YI=YT
6540 FOR J=0 TO 3
6550 GOSUB3560
6560 GOSUB3610
6570 NEXT J
6580 GOSUB3560
6590 GOSUB2920
6600 RETURN
6610 :
6620 :REM JATSZMAKILEPES FELTETELE
6630 JX=1-SGN(ME(0))
6640 RETURN
6650 :
6660 :REM LEPESEGYUTTES
6670 GOSUB3430
6680 GOSUB6090
6690 GOSUB6370
6700 GOSUB6620
6710 RETURN
6720 :
6730 :REM ALEPES
6740 AB=0
6750 GOSUB6660
6760 RETURN
6770 :
6780 :REM BLEPES
6790 AB=1
6800 GOSUB6660
6810 RETURN
6820 :
6830 :REM ODAJATEK
6840 :IF JX<>0 THEN6950
```

```
6850 GOSUB6730
6860 IF JX=0 THEN GOSUB6730
6870 GOTO6840
6880 :
6890 :REM VISSZAJATEK
6900 :IF JX<>0 THEN6950
6910 GOSUB6780
6920 IF JX=0 THEN GOSUB6730
6930 GOTO6900
6940 :
6950 :GOSUB1340
6960 RETURN
6970 :
6980 :REM SZAMITASOK
6990 IF PP(0)=PP(1) THEN NY(0)=NY(0)+1:NY(1)=NY(1)+1:AB=2:RETURN
7000 IF PP(0)>PP(1) THEN NY(0)=NY(0)+2:AB=0:RETURN
7010 NY(1)=NY(1)+2:AB=1
7020 RETURN
7030 :
7040 :REM UZENET
7050 TX$="NYERT: ▮"CHR$(65+AB)+"▮ JATEKOS."
7060 IF AB>1 THEN TX$="A VEGEREDMENY: EGAL."
7070 X=1:Y=21
7080 GOSUB1160
7090 TX$="                , ,"
7100 Y=20
7110 IF AB>1 THEN TX$="        ,      ,        ,"
7120 GOSUB1160
7130 RETURN
7140 :
7150 :REM X SZOVEG
7160 TX$="X - TOVABB"
7170 X=1:Y=23
7180 GOSUB1160
7190 TX$="           ,"
7200 X=1:Y=22
7210 GOSUB1160
7220 RETURN
7230 :
7240 :REM XY SZOVEG
7250 TX$="W - BEFEJEZES"
7260 X=22:Y=21
7270 GOSUB1160
7280 TX$="           ,"
7290 X=22:Y=20
7300 GOSUB1160
7310 TX$="X - TOVABB,    Y - A SOROZAT VEGE"
7320 X=1:Y=23
7330 GOSUB1160
7340 TX$="      ,                 ,"
7350 X=1:Y=22
7360 GOSUB1160
7370 RETURN
7380 :
7390 :GET Q$
7400 IF Q$<>"X" THEN7390
7410 Q$=""
7420 RETURN
7430 :
7440 :GET Q$
7450 IF Q$="X" THENRETURN
7460 IF Q$="Y" THEN SX=1:RETURN
7470 IF Q$<>"W" THEN7440
7480 PX=1:SX=1
7490 RETURN
7500 :
7510 :REM X-RE TOVABB
7520 GOSUB7150
7530 GOSUB7390
7540 GOSUB1340
7550 RETURN
7560 :
7570 :REM TOVABB V. SOROZAT VEGE
7580 GOSUB7240
7590 GOSUB7440
7600 GOSUB1340
7610 RETURN
7620 :
7630 :REM ODABEF & VISSZABEF
7640 GOSUB6980
7650 GOSUB2450
7660 GOSUB7040
7670 GOSUB7570
7680 GOSUB1280
7690 RETURN
7700 :
7710 :REM ODABEF
7720 X=1:Y=21:TX$="ODAVAGO VEGE. MOST: VISSZAVAGO!"
7730 GOSUB1160
7740 Y=20:TX$="          , ,  ,               , ,"
7750 GOSUB1160
7760 GOSUB7510
7770 GOSUB1280
7780 GOSUB1340
7790 RETURN
7800 :
7810 :REM ODAVAGO
7820 GOSUB3110
7830 GOSUB6830
7840 GOSUB7710
7850 RETURN
7860 :
7870 :REM VISSZAVAGO
7880 GOSUB3160
7890 GOSUB6890
7900 GOSUB7630
7910 RETURN
7920 :
7930 :REM JMAPAR
7940 GOSUB7810
7950 GOSUB7870
7960 RETURN
7970 :
7980 :REM JATSZMAPAR BEFEJEZES
7990 RETURN
8000 :
8010 :REM JMASOROZAT FORESZ
8020 :IF SX>0 THEN RETURN
8030 GOSUB3040
8040 GOSUB7930
8050 GOSUB7980
8060 GOTO8020
8070 :
8080 :REM JATSZMASOROZAT BEFEJEZESE
8090 RETURN
8100 :
8110 :REM KEZDOERTEK
8120 FOR JJ=0 TO 1
8130 IF SK(JJ)=0 THEN SK(JJ)=1
8140 SL(JJ)=SK(JJ)
8150 GOSUB2670
8160 NEXT JJ
8170 AB=0:Q=0:QQ=SK(0)
8180 RETURN
8190 :
8200 :REM STABIL SZOVEG
8210 TX$="HANY SZOMSZED? (NYILAK, RETURN, X)"
8220 X=1:Y=21:GOSUB1160
8230 TX$="     ,              ,"
8240 Y=20:GOSUB1160
8250 TX$="JATEKOS:":X=4:Y=23
```

```
8260 GOSUB1160
8270 TX$=" , ,":X=4:Y=22
8280 GOSUB1160
8290 RETURN
8300 :
8310 :REM ELO
8320 GOSUB1460
8330 GOSUB8630
8340 GOSUB1610
8350 Q$=""
8360 RETURN
8370 :
8380 :REM STABIL TORLESE
8390 GOSUB1340
8400 GOSUB1280
8410 RETURN
8420 :
8430 :REM BEVITEL
8440 :IF SK(0)<>SK(1) AND Q$="X" THEN RETURN
8450 :GET Q$
8460 IF Q$="▯" THEN GOSUB8680
8470 IF Q$="▯" THEN GOSUB8730
8480 IF Q$="▯" THEN Q=1:GOSUB1460
8490 IF Q$="▯" THEN Q=0:GOSUB1460
8500 IF Q$<>CHR$(13) THEN8440
8510 SL(AB)=QQ:SK(AB)=QQ:JJ=AB
8520 GOSUB2670
8530 GOTO8450
8540 :
8550 :REM JATSZMAPAR INPUT
8560 GOSUB8110
8570 GOSUB8200
8580 GOSUB8310
8590 GOSUB8430
8600 GOSUB8380
8610 RETURN
8620 :
8630 :REM QQ SZOMSZEDOK SZAMA
8640 TX$=CHR$(48+QQ)+"SZOM."
8650 X=14:Y=23
8660 GOSUB1160
8670 RETURN
8680 :REM FEL2
8690 IF Q=0 THEN GOSUB1740 :RETURN
8700 GOSUB8780
8710 RETURN
8720 :
8730 :REM LE2
8740 IF Q=0 THEN GOSUB1740 :RETURN
8750 GOSUB8840
8760 RETURN
8770 :
8780 :REM QQ--
8790 QQ=QQ-1
8800 IF QQ<1 THEN QQ=4
8810 GOSUB8630
8820 RETURN
8830 :
8840 :REM QQ++
8850 QQ=QQ+1
8860 IF QQ>4 THEN QQ=1
8870 GOSUB8630
8880 RETURN
8890 :
8900 :
8910 END

READY.
```

A MÁSODIK

Táblás játék

Tisztelt Commodore Újság!

A mellékelt lemezen küldöm — remélem, még időben — a stratégiai játék pályázatra a „Táblás játék" című programot, C64-es gépre. A játékot magam találtam ki, bár nem kétlem, hogy sok ilyen és ehhez hasonló játék létezik már.

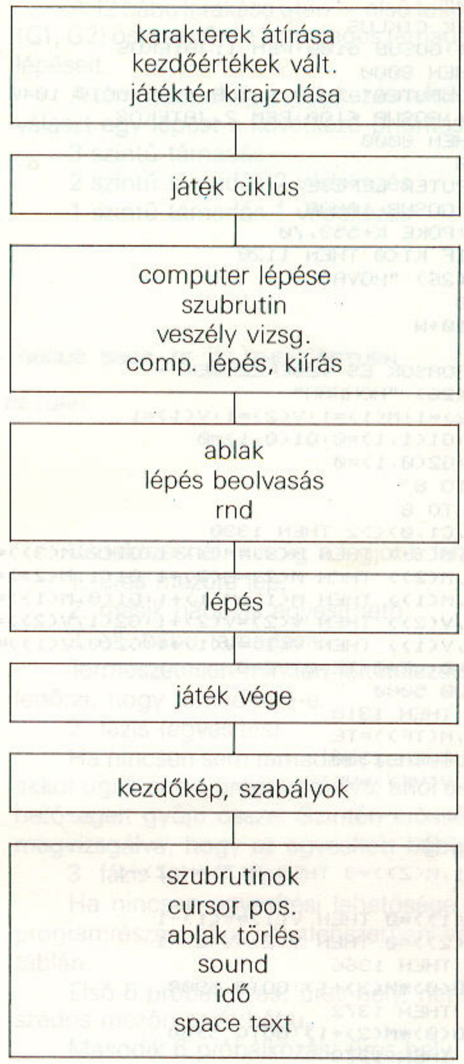
A játék szabálya:

A játék egy 8×8-as táblán folyik. A játékos játszhat egy másik játékos vagy a gép ellen is. Kezdetben mindkét játékosnak 64 bábuja van. A játék első részében a játékosok elhelyeznek 12-12 bábut. Itt még sem ütés, sem lépés nem lehetséges. Azt, hogy hány bábunk van még, a kép jobb oldalán lévő ablakokban folyamatosan figyelemmel tudjuk kísérni. Ezután már a játékos dönti el, hogy a táblán lévő bábuval lép vagy a tárból helyez új bábut a táblára. Tárból kilépés T (ütni nem lehet kilépéssel). A játékban az a pláne, hogy azonos színű bábukkal egymásra léphetünk, így erősebb bábukat hozva létre. Maximum 3 bábut egyesíthetünk. A kisebb fokozatú bábuval nagyobb fokozatút nem üthetünk. A játékosok felváltva lépnek, míg valamelyik fél bábui el nem fogynak. A cél természetesen az ellenfél teljes kiirtása.

```
0 REM ***********************
1 REM **  TABLAS JATEK     **
2 REM **ETNA  SOFTWARE'92**
3 REM **      IRTA:        **
4 REM ** BADENSZKY CSABA **
5 REM ***********************
10 POKE 56,116
20 POKE 54296,15
22 POKE 54277,7
24 POKE 54278,0
26 POKE 53272,21
100 REM - KARAKTER MASOLAS A ROM-BOL -
101 REM - AZ UJ KARAKTEREK BEOLVASASA -
110 PRINT"KIS TURELMET..."
120 POKE 56334,0:POKE 1,51
130 FOR Z=0 TO 463
140 :POKE 30720+Z,PEEK(53248+Z)
150 NEXT
160 FOR Z=0 TO 7
170 :POKE 31216+Z,PEEK(53760+Z)
180 :POKE 31224+Z,PEEK(53776+Z)
190 :POKE 31232+Z,PEEK(54144+Z)
200 :POKE 31240+Z,PEEK(54128+Z)
210 :POKE 31248+Z,PEEK(54120+Z)
220 :POKE 31256+Z,PEEK(54248+Z)
230 :POKE 31264+Z,PEEK(54104+Z)
240 :POKE 31272+Z,PEEK(54168+Z)
242 :POKE 31280+Z,PEEK(54528+Z)
250 NEXT
260 POKE 1,55:POKE 56334,1
270 FOR Z=0 TO 167
280 :READ A
290 :POKE 30936+Z,A
300 NEXT
310 FOR Z=0 TO 31
320 :READ A
```

```
330 :POKE 31184+Z,A
340 NEXT
342 FOR Z=0 TO 7
344 :READ A
346 :POKE 30720+Z,A
348 NEXT
350 DATA 85,85,85,85,85,85,85,85,
351 DATA170,170,170,170,170,170,170,170
352 DATA125,125,125,125,125,125,125,125
353 DATA255,87,87,255,255,213,213,255
354 DATA255,87,87,127,127,87,87,255
355 DATA0,0,0,0,0,0,0,0
356 DATA190,190,190,190,190,190,190,190
357 DATA255,171,171,255,255,234,234,255
358 DATA255,171,171,191,191,171,171,255
359 DATA85,85,84,87,87,95,95,93
360 DATA85,85,255,255,215,85,85,85
361 DATA85,85,85,213,213,245,245,117
362 DATA125,125,117,117,117,117,125,125
363 DATA125,125,93,93,93,93,125,125
364 DATA93,95,95,87,87,85,85,85
365 DATA85,85,85,215,255,255,85,85
366 DATA117,245,245,213,213,85,85,85
367 DATA170,170,170,171,171,175,175,174
368 DATA170,170,255,255,235,170,170,170
369 DATA170,170,170,234,234,250,250,186
370 DATA190,190,186,186,186,186,190,190
371 DATA190,190,174,174,174,174,190,190
372 DATA174,175,175,171,171,170,170,170
373 DATA170,170,170,235,255,255,170,170
374 DATA186,250,250,234,234,170,170,170
375 DATA0,24,60,60,126,126,126,60
376 DATA 36,37,38,39,27,40,41,42,43
377 DATA 44,45,46,47,28,58,59,60,61
380 REM ----------------------------------
400 DIM R(8,1)
402 FOR C1=0 TO 1
404 :FOR C2=0 TO 8
406 ::READ R(C2,C1)
408 :NEXT
410 NEXT
412 DIM T(8,8,1),G1(2,50),G2(1,50),M(3),V(2)
418 GOTO 9000
420 FOR C1=1 TO 8
422 :FOR C2=1 TO 8
424 ::T(C2,C1,0)=0:T(C2,C1,1)=0
426 :NEXT
428 NEXT
430 J1=64:J2=64:J3=30085:J4=30645
440 J5=64:J6=64:J7=4:J8=4:KT=12
444 RD=56324
450 K=29696:C=55296
500 POKE 56576,2 :POKE 53272,222
502 POKE 53270,31:POKE 648,116
504 POKE 53282,0 :POKE 53283,15
520 PRINT CHR$(147);
530 FOR Z=0 TO 7
540 :POKE K+Z*3+2,1+Z
550 :POKE C+Z*3+2,1
560 :POKE K+Z*120+80,49+Z
570 :POKE C+Z*120+80,1
580 NEXT
590 A$(0)="£££":A$(1)="[[["
600 POKE 646,8
610 FOR C1=0 TO 23
620 :PRINT CHR$(13) CHR$(29);
630 :FOR C2=0 TO 7
640 ::PRINTA$(-(((C1/3)AND1)=(C2AND1)));
650 :NEXT
660 NEXT
670 X=25:Y= 1:GOSUB 10000
680 GOSUB 6000
690 PRINT
692 PRINT TAB(25) "?->>>>>>>>>>>>>*":REM SHIFT+*
694 PRINT TAB(25) "?=            ?":REM 12 SPACE
696 PRINT TAB(25) "?=            ?"
698 PRINT TAB(25) "(>>>>>>>>>>>>>-":REM SHIFT+@ A SHIF
700 GOSUB 6000
710 X=26:Y=2:GOSUB 10000
720 PRINT CHR$(144) N1$
```

A következőkben a program felépítése és leírása következik.



A játéktér kirajzolása után a program meghívja a 6100-as adatbeolvasó rutint. Itt ellenőrzi a lépés helyességét és ugrik a 7000-es kirajzoló rutinra. Utána, ha nem gép ellen játszunk, újra meghívja a beolvasó rutint, ezúttal a 2. játékos számára. Ha a gép ellen játszunk, meghívja az 1100-as rutint, melynek végeredménye 4 szám, a lépés koordinátái. Ezt az 5900-as rutinnal kiírja az üzenetablakba. Innen jön a 6100-as rutin táblázatkezelő része és utána megint csak a 7000-es kirajzoló rutin.

Ha sokalljuk a 64-64 bábut, a 440, 430 sorokban a j1, j2, j5, j6 változókat átírhatjuk.

```
730 X=26:Y=16:GOSUB 10000
740 PRINT N2$
750 PRINT CHR$(5)
990 TI$="000000"
1000 REM - JATEK CIKLUS -
1010 J=1:N$=N1$:GOSUB 6100:REM 1.JATEKOS
1012 IF J6=0 THEN 8000
1020 IF N2$="COMPUTER" THEN GOSUB 1100:GOTO 1040
1030 J=2:N$=N2$:GOSUB 6100:REM 2.JATEKOS
1040 IF J5=0 THEN 8000
1090 GO TO 1000
1100 REM - COMPUTER LEPESE -
1102 X=26:Y=12:GOSUB 10000
1104 PRINT N2$:POKE K+553,70
1105 KT=KT-.5:IF KT<0 THEN 1120
1106 PRINT TAB(26) "HOVA"
1110 GOSUB 6500
1116 Z=9000+Q*10+W
1118 GOTO 5900
1120 REM - TAMADASOK ES VEDEKEZESEK -
1122 PRINT TAB(26) "HONNAN"
1130 M(3)=1:M(2)=1:M(1)=1:V(2)=1:V(1)=1
1134 G1(2,1)=0:G1(1,1)=0:G1(0,1)=0
1136 G2(1,1)=0:G2(0,1)=0
1150 FOR C1=1 TO 8
1154 :FOR C2=1 TO 8
1156 ::IF T(C2,C1,0)<>2 THEN 1330
1160 ::IF G1(2,M(3)) THEN M(3)=M(3)+1:G1(2,M(3))=0
1162 ::IF G1(1,M(2)) THEN M(2)=M(2)+1:G1(1,M(2))=0
1164 ::IF G1(0,M(1)) THEN M(1)=M(1)+1:G1(0,M(1))=0
1166 ::IF G2(1,V(2)) THEN V(2)=V(2)+1:G2(1,V(2))=0
1168 ::IF G2(0,V(1)) THEN V(1)=V(1)+1:G2(0,V(1))=0
1180 ::VV=1:TF=0:VF=0:TE=0:VE=0
1200 ::P=1:GOSUB 5000
1300 ::IF TF=0 THEN 1310
1304 ::G1(TF-1,M(TF))=TE
1310 ::IF VF=0 THEN 1330
1320 ::G2(VF-1,V(VF))=VE
1330 :NEXT
1332 NEXT
1340 FOR Z=1 TO 3
1344 :IF G1(Z-1,M(Z))=0 THEN M(Z)=M(Z)-1
1348 NEXT
1352 IF G2(0,V(1))=0 THEN V(1)=V(1)-1
1354 IF G2(1,V(2))=0 THEN V(2)=V(2)-1
1360 IF M(3)=0 THEN 1366
1362 Z=G1(2,RND(0)*M(3)+1):GOTO 5900
1366 IF M(2)=0 THEN 1372
1368 Z=G1(1,RND(0)*M(2)+1):GOTO 5900
1372 IF V(2)=0 THEN 1378
1374 Z=G2(1,RND(0)*V(2)+1):GOTO 5900
1378 IF M(1)=0 THEN 1384
1380 Z=G1(0,RND(0)*M(1)+1):GOTO 5900
1384 IF V(1)=0 THEN 1400
1386 Z=G2(0,RND(0)*V(1)+1):GOTO 5900
1400 REM - AZONOS BABUK EGYESITESE -
1410 V(2)=1:V(1)=1:P=3
1414 G2(1,1)=0:G2(0,1)=0
1420 FOR C1=1 TO 8
1422 :FOR C2=1 TO 8
1426 ::IF T(C2,C1,0)<>2 THEN 1450
1430 ::IF G2(1,V(2)) THEN V(2)=V(2)+1:G2(1,V(2))=0
1432 ::IF G2(0,V(1)) THEN V(1)=V(1)+1:G2(0,V(1))=0
1434 ::VF=0:VE=0
1440 ::GOSUB 5000
1442 ::IF VF=0 THEN 1450
1446 ::G2(VF-2,V(VF-1))=VE
1450 :NEXT
1452 NEXT
1460 IF G2(1,V(2))=0 THEN V(2)=V(2)-1
1462 IF G2(0,V(1))=0 THEN V(1)=V(1)-1
1470 IF V(2)=0 THEN 1480
1472 Z=G2(1,1+RND(0)*V(2)):GOTO 5900
1480 IF V(1)=0 THEN 1500
1484 Z=G2(0,1+RND(0)*V(1)):GOTO 5900
1500 REM - LEPES A TARBOL -
1510 IF J2=0 THEN 1600
1514 P=4:Z=1 : Z3=2:Z4=0
1516 FOR Z5=1 TO 6
1518 :GOSUB 6500
1520 :IF Z3=0 THEN Z4=1:Z5=6:GOTO 1540
1522 :IF Z3=1 THEN 1530
1524 :Z2=0:C2=Q:C1=W
1526 :GOSUB 5000
1527 :IF Z2=0 THEN 1540
1528 :Q=C2:W=C1
1530 :GOSUB 5800
1534 :IF Z2 THEN 1540
1538 :Z4=1:Z5=6
1540 NEXT
1544 IF Z4 THEN Z=9000+Q*10+W:GOTO 5900
1548 Z3=Z3-1
1550 GOTO 1516
1600 REM - LEPES -
1610 VF=0
1614 FOR C1=1 TO 8
1618 :FOR C2=1 TO 8
1622 ::IF T(C2,C1,0)<>2 THEN 1760
1626 ::VE=0:Z3=99
1630 ::FOR C3=1 TO 8
1634 :::FOR C4=1 TO 8
1638 ::::IF T(C4,C3,0)=0 THEN 1740
1642 ::::Z9=ABS(C2-C4)+ABS(C1-C3)
1646 ::::IF Z9>=Z3 OR Z9=0 THEN 1740
1648 ::::Z=T(C2,C1,1):Z8=T(C4,C3,1)
1650 ::::IF T(C4,C3,0)=1 AND Z8>Z THEN 1740
1654 ::::IF T(C4,C3,0)=2 AND Z+Z8>3 THEN 1740
1680 ::::Z4=SGN(C4-C2):Z5=SGN(C3-C1)
1684 ::::Q=C2+Z4:W=C1+Z5
1686 ::::IF T(Q,W,0) THEN 1696
1688 ::::GOSUB 5800
1692 ::::IF Z2=0 THEN 1730
1696 ::::Q=C2+Z4:W=C1
1702 ::::IF T(Q,W,0) THEN 1712
1704 ::::GOSUB 5800
1708 ::::IF Z2=0 THEN 1730
1712 ::::Q=C2:W=C1+Z5
1718 ::::IF T(Q,W,0) THEN 1740
1720 ::::GOSUB 5800
1724 ::::IF Z2 THEN 1740
1730 ::::Z3=Z9
1734 ::::VE=C2*1000+C1*100+Q*10+W
1740 :::NEXT
1742 ::NEXT
1750 ::IF VE=0 THEN 1760
1754 ::VF=VF+1
1758 ::G1(0,VF)=VE
1760 :NEXT
1764 NEXT
1766 IF VF=0 THEN 1800
1770 Z=G1(0,1+RND(0)*VF):GOTO 5900
1800 V(1)=1:V(2)=1:P=5
1804 G2(0,1)=0:G2(1,1)=0
1808 FOR C1=1 TO 8
1812 :FOR C2=1 TO 8
1816 ::IF T(C2,C1,0)<>2 THEN 1838
1820 ::IF G2(0,V(1)) THEN V(1)=V(1)+1:G2(0,V(1))=0
1824 ::IF G2(1,V(2)) THEN V(2)=V(2)+1:G2(1,V(2))=0
1828 ::VF=0:GOSUB 5000
1832 ::IF VF=0 THEN 1838
1834 ::G2(VF-1,V(VF))=VE
1838 :NEXT
1842 NEXT
1850 IF G2(0,V(1))=0 THEN V(1)=V(1)-1
1854 IF G2(1,V(2))=0 THEN V(2)=V(2)-1
1858 IF V(2)=0 THEN 1866
1862 Z=G2(1,1+RND(0)*V(2)):GOTO 5900
1866 IF V(1)=0 THEN RETURN
1870 Z=G2(0,1+RND(0)*V(1)):GOTO 5900
5000 REM -=( BABUK KORULVIZSGALASA )=-
5010 Q=C2:W=C1-1
5012 IF W<1 THEN 5016
5014 GOSUB 5080
5016 Q=C2:W=C1+1
5018 IF W>8 THEN 5022
5020 GOSUB 5080
5022 Q=C2-1:W=C1
5024 IF Q<1 THEN 5028
5026 GOSUB 5080
5028 Q=C2+1:W=C1
```

```
5030 IF Q>8 THEN 5034
5032 GOSUB 5080
5034 Q=C2-1:W=C1-1
5036 IF Q<1 OR W<1 THEN 5040
5038 GOSUB 5080
5040 Q=C2+1:W=C1+1
5042 IF Q>8 OR W>8 THEN 5046
5044 GOSUB 5080
5046 Q=C2-1:W=C1+1
5048 IF Q<1 OR W>8 THEN 5052
5050 GOSUB 5080
5052 Q=C2+1:W=C1-1
5054 IF Q>8 OR W<1 THEN 5058
5056 GOSUB 5080
5058 RETURN
5080 ON P GOTO 5100,5200,5300,5400,5500
5100 IF T(Q,W,0)<>1 THEN RETURN
5104 IF T(Q,W,1)>T(C2,C1,1) THEN 5150
5108 IF T(Q,W,1)<T(C2,C1,1) THEN Z=T(C2,C1,1):GOSUB 5800:IF Z2 THEN RETURN
5112 IF T(Q,W,1)<TF THEN RETURN
5116 IF T(Q,W,1)=TF AND (PEEK(RD)AND 1) THEN RETURN
5120 IF T(C2,C1,1)=T(Q,W,1) THEN VV=0:VF=0
5122 TF=T(Q,W,1)
5124 TE=C2*1000+C1*100+Q*10+W
5130 RETURN
5150 IF VV=0 THEN RETURN
5154 VV=0:Z3=0
5156 P=2:GOSUB 5000
5158 IF Z3=0 THEN 5162
5160 VF=T(C2,C1,1)
5162 RETURN
5200 IF T(Q,W,0)=0 THEN 5250
5204 IF T(Q,W,0)=2 THEN 5280
5210 IF T(Q,W,1)>=T(C2,C1,1)THEN RETURN
5214 Z=T(C2,C1,1)
5218 GOSUB 5800
5222 IF Z2 THEN RETURN
5224 IF Z3<3 THEN Z3=3:GOTO 5232
5228 IF PEEK(RD) AND 1 THEN RETURN
5232 VE=C2*1000+C1*100+Q*10+W
5236 RETURN
5250 Z=T(C2,C1,1)
5254 GOSUB 5800
5258 IF Z2 THEN RETURN
5262 IF Z3>1 THEN RETURN
5266 IF Z3<1 THEN Z3=1:GOTO 5274
5270 IF PEEK(RD) AND 1 THEN RETURN
5274 VE=C2*1000+C1*100+Q*10+W
5278 RETURN
5280 Z=T(C2,C1,1)+T(Q,W,1)
5282 IF Z>3 THEN RETURN
5284 GOSUB 5800
5286 IF Z2 THEN RETURN
5288 IF Z3=3 THEN RETURN
5290 IF Z3<2 THEN Z3=2:GOTO 5294
5292 IF PEEK(RD) AND 1 THEN RETURN
5294 VE=C2*1000+C1*100+Q*10+W
5296 RETURN
5300 IF T(Q,W,0)<>2 THEN RETURN
5310 Z=T(Q,W,1)+T(C2,C1,1)
5314 IF Z>3 THEN RETURN
5320 IF VF>Z THEN RETURN
5330 GOSUB 5800
5334 IF Z2 THEN RETURN
5338 IF VF<Z THEN VF=Z:GOTO 5346
5342 IF PEEK(RD) AND 1 THEN RETURN
5346 VE=C2*1000+C1*100+Q*10+W
5350 RETURN
5400 IF T(Q,W,0)=2 THEN Z2=1
5410 RETURN
5500 IF T(Q,W,0) THEN RETURN
5510 Z=T(C2,C1,1):GOSUB 5800
5520 IF Z2=0 THEN 5550
5522 IF VF=2 THEN RETURN
5524 IF VF=0 THEN VF=1:GOTO 5532
5528 IF PEEK(RD) AND 1 THEN RETURN
5532 VE=C2*1000+C1*100+Q*10+W
5536 RETURN
5550 IF VF<2 THEN VF=2:GOTO 5558
```

A gép lépése:

1. fázis (támadás, védekezés).

A 12 bábu lerakása után — első fokozatként — 2 tömbbe (G1, G2) összegyűjti a lehetséges támadások és védekezések lépéseit.

A támadásnak 3, a védekezésnek 2 fázisa van. Utána kiválaszt egy lépést a következő prioritási szint szerint.

3 szintű támadás
2 szintű támadás 2 védekezés
1 szintű támadás 1 védekezés

Védekezésnél mindig a legjobb lehetőséget választja:
1. üres mezőre lép,
2. másik bábuval egyesíthető,
3. ell. bábu támadása.

Természetesen minden feltételezett (rész-) lépésnél ellenőrzi, hogy támadható-e.

2. fázis (egyesítés).

Ha nincsen sem támadási, sem védekezési lehetősége, akkor ugrik erre a programrészre, ahol a bábuk egyesítési lehetőségeit gyűjti össze. Szintén erősségi fokozatokban és megvizsgálva, hogy az egyesített bábu nem támadható-e.

3. fázis (tárból lépés)

Ha nincsen egyesítési lehetősége, akkor ugrik erre a programrészre, ahol véletlenszerűen választ ki helyeket a táblán.

Első 6 próbálkozás: üres hely, nem támadható, szomszédos mezőn saját bábu.

Második 6 próbálkozás: üres hely, nem támadható.

Utána: üres hely, mindegy.

4. fázis (önálló lépés).

Ha a tár már üres, akkor jön ez a programrész, amely messze a legidőigényesebb az összes között. Itt minden egyes bábuval összehasonlítja az összes (távoli) többbit. Minden egyes bábuhoz kiválaszt egyet. Mindig a legközelebb lévőt, megvizsgálva, hogy ha ellenség, támadható-e, ha nem ellenség, egyesíthető-e. Itt is a vizsgált lépéseket egy tömbbe gyűjti össze, ahonnan a ciklus végén kiválaszt egyet.

Érdemes úgy játszani, hogy a végére (amikor a gépnek nincsenek bábui a tárban) viszonylag kevés bábu legyen a táblán, mert ez a programrész perceket is igénybe vehet. Érdemes lenne lefordítva kipróbálni, de sajnos nekem nincs fordítóprogramom, így erről nem tudok beszámolni.

A programból kilépni a STOP + Restore és a Poke648,4-gyel lehet.

A program kezelése egyszerű, mindig először az oszlop betűjét és a sor számát kéri. A beolvasó rész más bevitelt nem is enged. A lépésünket már a gép gondolkozása közben beírhatjuk. A gépben aktív action cartridge nem lehet.

A programhoz kellemes szórakozást kíván:

Badenszky Csaba

```
5554 IF PEEK(RO) AND 1 THEN RETURN
5558 VE=C2*1000+C1*100+Q*10+W
5562 RETURN
5800 REM - VESZELY VIZSGALAT -
5802 Z2=0
5804 E=Q:R=W-1
5806 IF R<1 THEN 5810
5808 GOSUB 5850:IF Z2 THEN RETURN
5810 E=Q:R=W+1
5812 IF R>8 THEN 5816
5814 GOSUB 5850:IF Z2 THEN RETURN
5816 E=Q-1:R=W
5818 IF E<1 THEN 5822
5820 GOSUB 5850:IF Z2 THEN RETURN
5822 E=Q+1:R=W
5824 IF E>8 THEN 5828
5826 GOSUB 5850:IF Z2 THEN RETURN
5828 E=Q-1:R=W-1
5830 IF E<1 OR R<1 THEN 5834
5832 GOSUB 5850:IF Z2 THEN RETURN
5834 E=Q+1:R=W+1
5836 IF E>8 OR R>8 THEN 5840
5838 GOSUB 5850:IF Z2 THEN RETURN
5840 E=Q-1:R=W+1
5842 IF E<1 OR R>8 THEN 5846
5844 GOSUB 5850:IF Z2 THEN RETURN
5846 E=Q+1:R=W-1
5848 IF E>8 OR R<1 THEN RETURN
5850 IF T(E,R,0)=1 AND T(E,R,1)>=Z THEN Z2=1
5852 RETURN
5900 REM - COMP. LEPES KIIRAS -
5902 Z2=K+553:FR=80
5904 IF Z>9000 THEN Q=20:Z=Z-9000:GOTO 5910
5906 Q=INT(Z/1000):Z=Z-Q*1000
5908 W=INT(Z/100) :Z=Z-W*100
5910 E=INT(Z/10)  :Z=Z-E*10
5912 R=Z:IF KT>=0 THEN 5934
5914 POKE Z2,Q:POKE Z2+1,70
5916 GOSUB 10040:GOSUB 10100
5918 IF Q=20 THEN 5924
5920 POKE Z2+1,W+48:POKE Z2+2,70
5922 GOSUB 10040:GOSUB 10100
5924 GOSUB 10040
5926 GOSUB 10030
5928 X=26:Y=12:GOSUB 10000
5930 PRINT N2$
5932 PRINT TAB(26) "HOVA"
5934 POKE Z2,70:GOSUB 10100
5936 POKE Z2,E:POKE Z2+1,70
5938 GOSUB 10040:GOSUB 10100
5940 POKE Z2+1,R+48:POKE Z2+2,70
5942 GOSUB 10040:GOSUB 10100
5944 GOSUB 10040
5946 GOSUB 10030
5948 J=2
5950 GOTO 6146
6000 :
6001 REM ---===( ABLAK )===---
6002 :
6010 PRINT TAB(25)"▙->>>>>>>>>>>>>*":REM SHIFT+*
6012 PRINT TAB(25)"?              ?"
6014 PRINT TAB(25)"->>>>>>>>>>>>>>-":REM SHIFT+D & SHIFT+E
6016 FOR Z=1 TO 5
6018 :PRINT TAB(25)"?▓▓▓▓▓▓▓▓▓▓▓▓▓▙?"
6020 NEXT
6022 PRINT TAB(25)"?▓▓▓▓▓         ▙?"
6024 PRINT TAB(25)"|>>>>>>>>>>>>>>-";:REM SHIFT+B & SHIFT+C
6026 RETURN
6100 :
6101 REM ---===( LEPES BEOLV. )===---
6102 :
6104 KT=KT-.5:IF KT>=0 THEN Q=20:GOTO 6122
6106 X=26:Y=12:GOSUB 10000
6108 PRINT N$
6112 PRINT TAB(26) "HONNAN"
6114 GOSUB 10050
6116 Q=ASC(A$)-64:W=ASC(A2$)-48
6118 GOSUB 10030
6122 X=26:Y=12:GOSUB 10000
6126 PRINT N$
6130 PRINT TAB(26) "HOVA"
6134 GOSUB 10050
6138 E=ASC(A$)-64:R=ASC(A2$)-48
6140 GOSUB 10030
6142 IF E=20 THEN KT=KT+.5:GOTO 6100
6146 IF NOT Q=20 THEN 6300
6150 IF J=1 AND J1=0 THEN 6100
6154 IF J=2 AND J2=0 THEN 6100
6158 IF NOT T(E,R,0)=0 THEN KT=KT+.5:GOTO 6100
6160 T(E,R,0)=J:T(E,R,1)=1
6162 IF J=2 THEN 6200
6166 J1=J1-1
6170 POKE J3,32
6174 J3=J3-1:J7=J7-1
6178 IF J7=0 THEN J7=12:J3=J3-28
6182 GOTO 6220
6200 J2=J2-1
6204 POKE J4,32
6208 J4=J4-1:J8=J8-1
6212 IF J8=0 THEN J8=12:J4=J4-28
6220 GOTO 7100
6300 IF NOT T(Q,W,0)=J THEN 6100
6304 IF Q=E AND W=R THEN 6100
6308 IF ABS(Q-E)>1 OR ABS(W-R)>1 THEN 6100
6312 IF T(Q,W,0)=T(E,R,0) THEN 6400
6316 IF NOT T(Q,W,1)>=T(E,R,1) THEN 6100
6317 IF T(E,R,0) THEN K1=1
6318 IF J=1 THEN J6=J6-T(E,R,1)
6320 IF J=2 THEN J5=J5-T(E,R,1)
6322 T(E,R,0)=T(Q,W,0):T(E,R,1)=T(Q,W,1)
6324 T(Q,W,0)=0:T(Q,W,1)=0
6340 GOTO 7000
6400 IF T(Q,W,1)+T(E,R,1)>3 THEN 6100
6410 T(E,R,1)=T(E,R,1)+T(Q,W,1)
6420 T(Q,W,0)=0:T(Q,W,1)=0
6430 K2=1
6460 GOTO 7000
6500 REM ---===( RND )===---
6510 Q=1+INT(RND(0)*8)
6512 W=1+INT(RND(0)*8)
6516 IF T(Q,W,0) THEN 6510
6518 RETURN
7000 :
7001 REM ---===( LEPES )===---
7002 :
7010 REM - TORLES -
7012 FR=30:GOSUB 10040
7020 Z=K+1+(Q-1)*3+(W-1)*120+40
7024 Z2=27+(Q+W AND 1)
7030 FOR C1=0 TO 2
7040 :FOR C2=0 TO 2
7044 ::POKE Z+C1*40+C2,Z2
7048 :NEXT
7052 NEXT
7100 REM - RAJZOLAS -
7102 FR=20:GOSUB 10040
7110 Z=1+(E-1)*3+(R-1)*120+40
```

```
7114 Z2=(E+R AND 1)
7116 Z3=J+8
7120 FOR C1=0 TO 2
7124 :FOR C2=0 TO 2
7130 ::POKEK+Z+C1*40+C2,R(C1*3+C2,Z2)
7132 ::POKEC+Z+C1*40+C2,Z3
7134 :NEXT
7138 NEXT
7140 POKE K+Z+41,28+T(E,R,1)+4*Z2
7142 IF K1 THEN K1=0:FR=12:GOSUB 10044
7144 IF K2 THEN K2=0:FOR Z=10 TO 40 STEP 5:FR=Z:GOSUB 10040:NEXT
7150 RETURN
8000 POKE 56576,3:POKE 53272,21
8010 POKE 648,4:POKE 53270,200
8030 A$=N$+" VOLT AZ UGYESEBB"
8040 PRINT "[][][][]" TAB(13) "VEGE A JATEKNAK"
8050 PRINT:PRINT
8060 PRINT TAB((40-LEN(A$))/2) "■" A$
8070 A$=""
8080 IF LEFT$(TI$,2)="00" THEN 8100
8090 A$=LEFT$(TI$,2)+" ORA "
8100 A$=A$+MID$(TI$,3,2)+" PERCET JATSZOTTATOK"
8110 PRINT:PRINT:PRINT
8120 PRINT TAB((40-LEN(A$))/2) "■" A$
8130 PRINT"[][][][]"
8140 POKE 198,0:K=1:GOSUB 10110
9000 PRINT CHR$(147) CHR$(8)
9010 POKE 53280,6:POKE 53281,6
9012 K=1:POKE 198,0
9014 X=14:Y=2:GOSUB 10000
9020 PRINT CHR$(5)  "TABLAS JATEK"
9030 PRINT TAB(14) CHR$(144)"────────────────":REM 16 SHIFT+E
9040 PRINT
9050 PRINT TAB(10) CHR$(158)"IRTA:BADENSZKY CSABA"
9060 PRINT:PRINT
9062 PRINT CHR$(155) "SZABALYOK:EGY 8X8 MEZON LEHET FELVALTVA"
9064 PRINT "LEPEGETNI.MINDKET JATEKOSNAK 64 BABUJA"
9066 PRINT "VAN.A JATEKOS DONTI EL,HOGY A TARBOL,"
9068 PRINT "VAGY MAR EGY PALYAN LEVO BABUVAL LEP."
9069 PRINT "A PALYAN AZ AZONOS BABUKKAL EGYMASRA "
9070 PRINT "LEPHETUNK,NOVELVE AZ EROSSEGI FOKOZATOT"
9071 PRINT "MAXIMUM 3-IG.HA A TARBOL AKARUNK LEPNI"
9072 PRINT "A 'HONNAN' KERDESRE 'T' KELL GEPELNI."
9073 PRINT "A JATEK KEZDETEN A JATEKOSOK FELRAKNAK"
9074 PRINT "12-12 BABUT.UTANNA MINDEN A LEIRTAK"
9076 PRINT "SZERINT MEGY.GYOZTES:AKI LEUTI AZ"
9078 PRINT "ELLENFEL OSSZES BABUJAT."
9080 PRINT:PRINT CHR$(5)
9090 GOSUB 10110
9200 INPUT " 1.JATEKOS";N1$
9210 IF LEN(N1$)>12 THEN N1$=LEFT$(N1$,12)
9220 INPUT " 2.JATEKOS  COMPUTER[][][][][][][][][][]";N2$ : REM 10 <CRSR BALRA>
9230 IF LEN(N2$)>12 THEN N2$=LEFT$(N2$,12)
9250 GOTO 420
10000 REM ---===( CURSOR POS. )===---
10010 POKE 211,X:POKE 214,Y:SYS 58640
10020 RETURN
10030 REM ---===( ABLAK TORL. )===---
10032 FOR Z=30202 TO 30213
10034 : POKE Z,32:POKE Z+40,32
10036 NEXT
10038 RETURN
10040 REM ---===( SOUND )===---
10042 POKE 54276,0:POKE 54276,17
10043 GOTO 10046
10044 POKE 54276,0:POKE 54276,129
10046 POKE 54273,FR
10048 RETURN
10050 REM ---===( BILL. LEKERD. )===---
10052 Z=K+553:FR=80
10054 POKE Z,70
10056 WAIT 198,255:GET A$
10057 IF A$<"A" OR A$>"H" AND A$<>"T" THEN 10056
10058 POKE Z,ASC(A$) AND 63
10060 POKE Z+1,70
10062 GOSUB 10040
10064 WAIT 198,255:GET A2$
10066 IF ASC(A2$)=20 THEN POKE Z+1,32:GOSUB 10040:GOTO 10054
10070 IF ASC(A2$)=13 AND A$="T" THEN 10090
10072 IF A2$<"1" OR A2$>"8" THEN 10064
10073 GOSUB 10040
10074 POKE Z+1,ASC(A2$)
10078 POKE Z+2,70
10080 WAIT 198,255:GET A3$
10084 IF ASC(A3$)=20 THEN POKE Z+2,32:GOTO 10060
10086 IF ASC(A3$)<>13 THEN 10080
10090 GOSUB 10040
10099 RETURN
10100 REM ---===( IDO )===---
10104 FOR Z=1 TO 220:NEXT
10108 RETURN
10110 REM ---===( SPACE TEXT )===---
10112 K=1-K:A$="<<< SPACE >>>"
10114 PRINT CHR$(18+K*128) CHR$(145);
10116 PRINT TAB(14) A$
10118 GET K$
10120 FOR Z=1 TO 80:NEXT
10122 IF NOT K$=" " THEN 10112
10128 RETURN
READY.
```

A HARMADIK

SEDA

Tisztelt Zsűri!

Az alábbiakban mellékelt SEDA nevű programommal szeretném benevezni az Önök BASIC programnyelvű stratégiai játékának megírására kiírt pályázatra.

A program a talán nem mindenki által ismert SZENEGÁLI DÁMAJÁTÉK-ot dolgozza fel.

A játék lényege a következő:

A játékot két játékos játszhatja, akik felváltva léphetnek egyet egy szomszédos üres mezőre jobbra, balra vagy az ellenfél térfelének irányába (1. ábra). A cél a másik bábuinak kiütése. Ha a lépő bábu és az üres hely között az ellenfél bábuja van, úgy az kiüthető, ezzel egy pont és egy üres hely szerezhető (2. ábra). Ha bábunkkal áttérünk az ellenfél térfelének legszélére, úgy azt elveszítjük.

A program indítása után megjelenik a kezdőkép, ahol kérhetjük az F1 billentyűvel a játék pontos leírását. Ha erre nem tartunk igényt, úgy egy akármilyen más billentyű lenyomására megjelenik a játéktér. Ennek bal felső részén az eddig szerzett pontok, középen pedig a 6×5-ös játékmező látható.

A játékot mi kezdjük (a játékos). A legelső mezőben megjelenik egy világos négyzet, amivel – a kurzormozgató billentyűket igénybe véve – kiválaszthatjuk a lépendő figurát. Ekkor az F1 billentyűt megnyomva balra (ha ez az irány megengedett), vagy az F3-mal lefelé (ha ez lehetséges), vagy az F5 billentyűvel jobbra (szintén, ha lehetséges) léphetünk. (Mindig arra a figurára kell ráállni, amellyel lépni szeretnénk!)

A mi lépésünk után a számítógép lép, majd megint mi következünk mindaddig, míg a gép vagy mi el nem érünk 14 pontot, azaz elveszíti valamelyik játékos az összes bábuját.

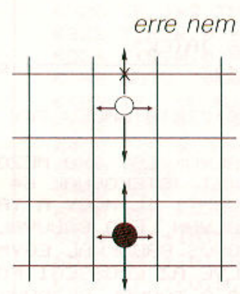
A gép értékelése után egy billentyű lenyomására újra kezdhetjük a játékot.

A program stratégiája a következő:

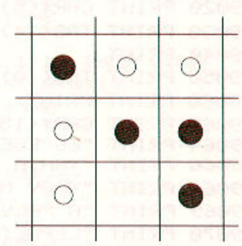
Miután mi léptünk, a gép végignézi az összes mezőt, és ha talál egy olyan kombinációt, ahol a saját bábuja és az üres hely között a mi bábunk van, valamint ha a haladási irány a három megengedett irány egyike, úgy lép egyet és kiüti a bábunkat. Ha nem létezik ilyen helyzet, úgy mégegyszer végigmegy a pályán, és keres egy olyan egyszerű lépést, amit léphet (és lépi is), majd ismét megjelenik a kis világos négyzet, ezzel jelezve, hogy mi következünk.

Nos, azt hiszem, ennél többet nem írhatok, a program önmagáért beszél. A gép egyébként nagyon nehezen győzhető le (10-ből ha háromszor), de sok sikert kívánok!

Sasvári Gábor
7635 Pécs, Bagoly u. 52.



1. ábra



2. ábra

```
10 REM * * * SZENEGALI DAMAJATEK * * *
11 REM
12 REM
13 REM             SEDA
14 REM
15 REM              BY
16 REM
17 REM        SASVARI GABOR
18 REM
19 REM
20 REM * A KEZDO ERTEKEK BEALLITASA *
21 PO1=0:P22=0:DIM HX(30),HY(30),A(7,8),D$(20):PC=0
30 GOSUB 7000 :REM * * * KEZDOKEP * * *
31 IF A$="■" THEN PC=1:
50 A$(1)="■■■■■":A$(2)="■■■■■":A$(3)="    ■■■    ":B$="■■    ■■■    ■"
51 FOR I=1 TO 2:FOR U=1 TO 6:A(I,U)=1:A(I+3,U)=2:NEXT:NEXT
52 FOR I=1 TO 6:READK:A(3,I)=K:NEXT:DATA 1,1,3,3,2,2
53 IF PC=1 THEN GOSUB 5000
97 REM
98 REM
99 REM * * * KEZDOKEP * * *
100 PRINT"■■":POKE 53280,0:POKE 53281,0:
110 PRINTTAB(11)"┌───┬───┬───┬───┬───┬───┐"
111 FOR I=1 TO 5:
112 PRINTTAB(11)"│   │   │   │   │   │   │"
113 PRINTTAB(11)"│   │   │   │   │   │   │"
114 IF I<>5 THEN  PRINTTAB(11)"├───┼───┼───┼───┼───┼───┤"
115 NEXT
116 PRINTTAB(11)"└───┴───┴───┴───┴───┴───┘"
118 PRINT"■■                    ■  ■  ■  ■■■ ■■"
120 PRINT"                      ■■ ■■  ■■ ■ ■   ■■"
122 PRINT"                      ■   ■   ■  ■■ ■ ■"
```

```
124 PRINT"■■■■ LEPESEK:■■■■■■■■■■■■■■■■■■■■■PONTSZAM:"
125 PRINT"■■BALRA- F1■■■■■■■■■■■■■■■■■■■■■UJATEKOS"
126 PRINT"■■LEFELE-F3
128 PRINT"■■JOBBRA-F5■■■■■■■■■■■■■■■■■■■■■SZAMITOGEP"
158 GOSUB 500
159 GOSUB 1000:GOSUB 2000:LEP=LEP+1
169 IF P0<14 AND P2<14 THEN 159
170 PRINT"■"
180 GOSUB 8000
497 REM
498 REM
499 REM *** A BABUK FELRAJZOLASA ***
500 PRINT"■■"
501 FOR I=1 TO 6:IF A(1,I)=2 THEN P0=P0+1:A(1,I)=3
502 IF A(5,I)=1 THEN P2=P2+1:A(5,I)=3
503 NEXT
521 FOR I=1 TO 5:FOR U=1 TO 6:PRINTTAB(9+U*3)A$(A(I,U))"■";:NEXT:PRINT
522 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"P0
523 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"P2
532 RETURN
998 REM
999 REM
1000 REM *** A LEPES KIVALASZTASA ***
1001 X=1:Y=1:L=0:K=0
1050 PRINT"■■":IF L>1 THEN FOR I=1 TO L-1:PRINT"■■":NEXT
1051 PRINTTAB(9+3*K)A$(A(L,K))
1052 PRINT"■■":IF Y>1 THEN FOR I=1 TO Y-1:PRINT"■■":NEXT
1053 PRINTTAB(9+3*X)B$
1060 GET A$:IF A$="" THEN 1060
1061 IF A$="■" AND Y<5   THEN K=X:L=Y:Y=Y+1
1062 IF A$="■" AND X<6   THEN K=X:L=Y:X=X+1
1063 IF A$="■" AND X>1   THEN K=X:L=Y:X=X-1
1064 IF A$="■" AND Y>1   THEN K=X:L=Y:Y=Y-1
1065 IF A$="■" OR A$="■" OR A$="■"  THEN GOTO 3000
1090 GOTO 1050
1998 REM
1999 REM
2000 REM * * * A GEP LEP * * *
2001 FOR I=1 TO 30:HY(I)=0:HX(I)=0:NEXT:S=1
2010 FOR I=1 TO 5:FOR U=1 TO 6:IF A(I,U)=3 THEN HY(S)=I:HX(S)=U:S=S+1
2011 NEXT:NEXT
2030 FOR I=S-1 TO 1 STEP -1
2040 IF A(HY(I),HX(I)+2)=2 AND A(HY(I),HX(I)+1)=1 THEN 2080
2042 IF A(HY(I)+2,HX(I))=2 AND A(HY(I)+1,HX(I))=1 THEN 2081
2043 IF HX(I)<2 THEN 2050
2044 IF A(HY(I),HX(I)-2)=2 AND A(HY(I),HX(I)-1)=1 THEN 2082
2050 NEXT:GOTO 2200
2080 A(HY(I),HX(I))=2:A(HY(I),HX(I)+1)=3:A(HY(I),HX(I)+2)=3:I=1:GOTO 2100
2081 A(HY(I),HX(I))=2:A(HY(I)+1,HX(I))=3:A(HY(I)+2,HX(I))=3:I=1:GOTO 2100
2082 A(HY(I),HX(I))=2:A(HY(I),HX(I)-1)=3:A(HY(I),HX(I)-2)=3:I=1:GOTO 2100
2100 NEXT:P2=P2+1:GOSUB 500:RETURN
2200 FOR I=S-1 TO 1 STEP -1
2201 IF A(HY(I),HX(I)-1)=2THEN:A(HY(I),HX(I))=2:A(HY(I),HX(I)-1)=3:I=1:GOTO2220
2202 IF A(HY(I)+1,HX(I))=2THENA(HY(I),HX(I))=2:A(HY(I)+1,HX(I))=3:I=1:GOTO2220
2203 IF A(HY(I),HX(I)+1)=2THEN:A(HY(I),HX(I))=2:A(HY(I),HX(I)+1)=3:I=1:GOTO2220
2220 NEXT:GOSUB 500:RETURN
2997 REM
2998 REM
2999 REM  * * * A LEPES * * *
3000 IF A(Y,X)=2 OR A(Y,X)=3 THEN 1050
3001 IF A$="■" OR A$="■" THEN 3100
3002 IF A(Y+2,X)=3 AND A(Y+1,X)=2 THEN 3010
3003 IF A(Y+1,X)<>3 THEN 1050
3004 GOSUB 3900
3005 A(Y,X)=3:A(Y+1,X)=1:GOTO 3030
3010 A(Y+1,X)=3:A(Y,X)=3:A(Y+2,X)=1:P0=P0+1
3030 GOSUB 500:RETURN
3100 IF A$="■" THEN 3200
3101 IF A(Y,X+2)=3 AND A(Y,X+1)=2 THEN 3110
3103 IF A(Y,X+1)<>3 THEN 1050
3104 GOSUB 3900
3105 A(Y,X)=3:A(Y,X+1)=1:GOTO 3130
3110 A(Y,X+1)=3:A(Y,X)=3:A(Y,X+2)=1:P0=P0+1
3130 GOSUB 500:RETURN
3200 IF X<2 THEN 3203
3201 IF A(Y,X-2)=3 AND A(Y,X-1)=2 THEN 3210
3203 IF A(Y,X-1)<>3 THEN 1050
3204 GOSUB 3900
3205 A(Y,X)=3:A(Y,X-1)=1:GOTO 3230
3210 A(Y,X-1)=3:A(Y,X)=3:A(Y,X-2)=1:P0=P0+1
```

```
3230 GOSUB 500:RETURN
3900 PRINT"▒▒":IF Y>1 THEN FOR I=1 TO Y-1:PRINT"▒▒":NEXT
3901 PRINTTAB(9+3*X)A$(A(Y,X)):RETURN
5000 REM
5001 REM
5002 REM  * * * JATEKLEIRAS  * * *
5003 PRINT"▒▒":POKE 53280,0:POKE 53281,0
5004 PRINTCHR$(14):SS=54272:POKE SS+24,15:POKE SS+5,9:POKE SS+6,0:
5005 FOR I=1 TO 20:READ D$(I):NEXT:POKE SS+1,50
5006 FOR I=1 TO 20:PRINTTAB((40-LEN(D$(I)))/2);
5007 FOR U=1 TO LEN(D$(I)):POKE SS+4,33:POKE SS+4,16:PRINTMID$(D$(I),U,1);
5008 FOR X=1 TO 30:NEXT
5018 NEXT
5019 PRINT
5020 NEXT
5021 GET A$:IF A$="" THEN 5021
5022 PRINT"▒":FOR I=1 TO 21:PRINT"
5023 FOR X=1 TO 50:NEXT
5025 NEXT:PRINTCHR$(142)
5026 RETURN
6000 DATA "▒▒▒/▒▒L\ ▒\▒ ▒▒▒","
6001 DATA "▒ATEKLEIRAS"
6002 DATA ""
6003 DATA " ▒ ▒ZENEGALI ▒AMAJATEK A  ▒\\\.SZ  KOR-"
6004 DATA "NYEKEN KELETKEZETT KETSZEMELYES TARSAS-"
6005 DATA "JATEK.▒ JATEK LENYEGE A KOVETKEZO:      "
6006 DATA "\IND A KET  JATEKOS  14-14 BABUVAL REN-"
6007 DATA "DELKEZIK.▒ CEL:A MASIK BABUJAINAK A KI-"
6008 DATA "UTESE.▒Z EPPEN  SORON KOVETKEZO JATEKOS"
6009 DATA "CSAK JOBBRA,BALRA,ES AZ ELLENFEL TERFE-"
6010 DATA "LE IRANYABA LEPHET.▒GYSZERRE CSAK EGYET"
6011 DATA "LEHET LEPNI EGY SZOMSZEDOS URES MEZORE."
6012 DATA " ▒A A LEPO BABU,ES AZ URES  MEZO  KOZOTT"
6013 DATA "AZ ELLENFEL BABUJA ALL,UGY AZ KIUTHETO,"
6014 DATA "S EZZEL PONT ES EGY URES HELY SZEREZHE-"
6015 DATA "TO. ▒A A BABUNK A SZEMKOZTI  OLDALRA AT-"
6016 DATA "ER,UGY AZT ELVESZITJUK.              ",""
6017 DATA "▒O SZORAKOZAST KIVANOK!"
7000 REM
7001 REM
7002 REM  * * *  KEZDOKEP  * * *
7003 PRINT"▒▒":
7004 POKE 53280,0:POKE 53281,0
7020 E$(1)="▒     ▒▒▒▒▒▒   ▒▒▒▒▒▒   ▒▒▒▒▒  ▒"
7022 E$(2)="▒▒▒  ▒▒▒▒▒▒    ▒▒▒▒▒  ▒  ▒▒▒▒  ▒ "
7024 E$(3)="▒      ▒▒▒▒▒    ▒▒▒▒  ▒ ▒▒▒▒▒▒  ▒  "
7030 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒              F1 - JATEKLEIRAS";
7031 PRINT"▒▒▒  EGYEB MAS BILLENTYU A JATEK INDITASA"
7039 N$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒":N$=N$+N$:N$=N$+N$+N$:N$=N$+N$
7040 FOR K=1 TO 250
7041 PRINTMID$(N$,K,1);
7049 PRINT"▒▒▒▒▒":FOR I=1 TO 3:PRINTTAB(6)E$(I):PRINT:NEXT
7050 PRINTMID$(N$,K+4,1);:GET A$:IF A$<>"" THEN RETURN
7051 PRINT"▒▒▒▒▒▒":FOR I=1 TO 3:PRINTTAB(5)E$(I):PRINT:NEXT
7060 NEXT
7061 GOTO 7040
8000 REM
8001 REM
8002 REM  * * * G A M E  O V E R * * *
8004 V$(1)="▒   ▒   ▒ ▒ ▒   ▒▒▒▒  ▒ ▒ ▒  ▒  "
8005 V$(2)="▒ ▒▒▒▒ ▒ ▒   ▒ ▒▒▒▒▒▒ ▒ ▒ ▒ ▒ ▒▒▒▒ ▒ "
8006 V$(3)="▒ ▒▒▒▒    ▒ ▒ ▒  ▒▒▒▒▒ ▒ ▒ ▒ ▒  ▒▒▒  ▒"
8007 V$(4)="▒ ▒ ▒ ▒ ▒ ▒ ▒ ▒▒▒▒▒▒ ▒ ▒ ▒ ▒ ▒▒▒▒ ▒ "
8008 V$(5)="▒   ▒ ▒ ▒ ▒ ▒   ▒▒▒▒   ▒▒▒ ▒▒▒  ▒ ▒ "
8015 FOR U=1 TO 10:POKE 646,4:IF U/2=INT(U/2) THEN POKE 646,1
8016 FOR X=1 TO 150:NEXT
8019 PRINT"▒▒▒":FOR I=1 TO 5:PRINTTAB(4)V$(I):NEXT
8020 NEXT
8022 PRINT"▒▒▒▒":FOR I=1 TO 3:PRINTTAB(4)"▒▒
8023 NEXT
8024 PRINT"▒▒▒":PRINTTAB(4)"▒/────────────────────────────────\"
8025 FOR I=1 TO 3:PRINTTAB(4)"|"TAB(36)"|":NEXT
8026 PRINTTAB(4)"\────────────────────────────────/"
8027 K$="  SAJNOS EN NYERTEM!  "
8028 IF PO=14THEN K$="GRATULALOK,MAGA NYERT!"
8029 PRINT"▒▒▒▒▒▒▒"TAB(10)"▒▒"K$
8030 GET A$:IF A$="" THEN 8030
8040 RUN

READY.
```

Képernyővarázslat — a TED

Ötödik rész: A hang programozása

Be kell valljam, félve ültem le a gép elé, mikor ezt a cikket írni kezdtem. Sok emberben ugyanis az a meggyőződés él, hogy a PLUS/4 hangja borzasztó. Ehhez nagyban hozzájárul a COMMODORE cég előző — joggal sikeres — gépe, a C-64. Ez a gép a maga kategóriájában csodálatos zenéket produkált, melyekhez a PLUS/4 zenék jó része igencsak gyengécske. Hogy a helyzetet tovább rontsák, megjelentek olyam ,,kitűnő" programozók, akik (C-64-ről átírt és egyéb) programjaikba ugyancsak gyengén sikerült ,,3 szólamú" zenéket raktak (a három C-64 szólamot a PLUS/4 két csatornáján gyorsan váltogatják), illetve újabban ,,digitkonvertereket" használnak, és mindkét módszerrel a köcsögdudára megtévesztésig hasonlító zajokat állítottak elő. A sok szidás után azonban szólnom kell az ellenpéldákról is: ügyes variálásokkal (melyekről majd később szólok) igen jó zenéket lehet előállítani. Erre példa a Nest of fleas, A borgok kincse, a Bard's Tale 3., a Bongo... Különben is, hasonlítsunk össze egy átlag PLUS/4 és egy átlag IBM PC zenét! A PLUS/4-ről legalább le lehet venni a hangot...

Nos, ennyi üres beszéd után nézzük meg, mi is történik, ha zenél a gép! A BASIC parancsok (VOL, SOUND) már bizonyára mindenkinek ismerősek. Gépi kódban ahhoz, hogy egy hang megszólaljon, először be kell állítani a frekvenciát, majd a hangerőt, engedélyezni kell a négyszögjelgenerátor működését, engedélyezni a csatornát, és az esetleges fehérzajt. Ez most lehet, hogy elsőre kicsit bonyolultnak tűnik, de ez egy hang esetén maximum három regiszter írását jelenti. Ebből kettő a hangmagasságot állítja, és egy látja el az összes többi funkciót. Nézzük a regiszterek listáját!

$FF0E — 1. hanggenerátor frekvencia alsó bájt
$FF0F — 2. hanggenerátor frekvencia alsó bájt
$FF10 — 2. hanggenerátor frekvencia felső bájt, az 1. és 0. biten
$FF11 — hangvezérlő regiszter
b.7 — 0/1 négyszögjel/analóg jel
b.6 – 1 = 2. generátorzaj engedélyezés
b.5 – 1 = 2. generátorhang engedélyezett
b.4 – 1 = 1. generátorhang engedélyezett
b.3 – 0 — hangerő (0–8-ig)
$FF12 — 1. hanggenerátor frekvencia felső két bit az 1. és 0. biten.

Mint azt már eddig is tudtuk, két generátorral rendelkezünk, melyek csak négyszögjelet tudnak szolgáltatni. A frekvenciát 10 biten programozhatjuk 6 oktáv terjedelemben (legalábbis a zenei hangokat). A hangterjedelem kiszámítható az alábbi képletből : A gépbe beírandó érték = 1024 — (c/f), ahol ,,c" egy konstans, PAL rendszer esetén 111840.45 (NTSC esetén 111860.781), f pedig a frekvencia Hz-ben. Az inverz képlet: f = c/(1024 — gépi érték). A gépbe beírható érték 10 biten 0-tól 2↑10—1-ig, azaz 1023-ig terjedhet. Minél magasabb a hang, annál magasabb az érték, de 1023 beírása mély hangot eredményez. Tehát a hangterjedelem c/1024-től (109.2 Hz) c/2-ig (55920 Hz, magasan az ember által hallható 16000—20000 Hz fölött). A hangerőt elméletileg négy biten programozhatjuk, de 8-nál nagyobb beírt érték nem jár hangerőnövekedéssel. Mint az az $FF11 regiszter leírásából látszik, a két generátor hangereje közös. Egy hanggenerátor akkor szólal meg, ha be van kapcsolva. Ezt külön-külön vezérelhetjük. Ha a kettes generátoron zajt kívánunk megszólaltatni, a zaj bitet 1-be kell vinni. Ha a zaj és négyszögjel is be van kapcsolva, a négyszögjel fog szólni. Próbáljuk ki az 1. listán látható programot! Az ígért trükkök: 2 szólam + dob megszólaltatható úgy, hogy a 2. generátoron futó szólamot dob idején kikapcsoljuk, és utána vissza. Ezt úgy is megtehetjük, hogy mind a két szólamot szüneteltetjük a dob megszólalásakor. A hangerővel érdemes játszani: ha a lejátszórutin ügyesen állítgatja, olyasmit kapunk, mint a burkológörbe állítgatásakor C-64-en (így működik az említett néhány jó zene). Sajnos, ezzel nagyjából kimerítettük a hanggenerátorok négyszögjel és zaj üzemmódjaival elérhető hatásokat. Egyéni ötletek persze jöhetnek még, például a zene időzítésére felhasználhatjuk a beépített megszakítórutint is, de ez gyakran túl nagy megkötéseket jelent. A $04FC/04FE cím az 1. generátor időzítője, a $4FD/04FF a második. Használatuk: bekapcsoljuk a hangokat, ahogy eddig, majd beírjuk az időzítőbe a hang hosszának kettes komplemensét (ugyanis az időzítőket a megszakítórutin növeli, és ha elérik a nullát, akkor kapcsol ki) 1/60 másodpercben. Ha lejár, a megszakítórutin a generátort automatikusan kikapcsolja (2. lista).

És most rátérnék az egyik kedvenc témámra: a hangdigitalizálásra. Ha szerényebb minőséggel is megelégszünk, nem kell hozzá nagy felszerelés, csak egy PLUS/4 és egy VC1531-es mágnesszalag drive egység, magyarul magnó (DATASETTE) na és persze egy program. Ezzel csak 1-bites digit lehet elérni, de ingyen van... Mi teszi lehetővé a hangdigitalizálást? Az $FF11 egyetlen ismeretlen bitje, a 7. bit. Ha ide egyet írunk, a hangerővel vezérelhető úgynevezett ,,analóg" jelet kapunk a kimeneten, a beépített négyszögjelgenerátor kikapcsol. Érdekesség, hogy nyolc helyett 12(!) hangerőt használhatunk. Ha ugyanis egy vagy két generátort használunk, a második esetben nagyobb változás tapasztalható az analóg kimeneten (tehát a ,,nulla" szint és két generátor 8-as hangerőn nagyobb feszültségkülönbséget okoz a hangkimeneten, mint egy). Így elméletileg 8*2=16 hangerőnk lenne, de ebből néhány egybeesik. Sajnálatos módon nem adhatom meg a hangerők sorrendjét, mert azokat nem én találtam ki, és nem szakkönyvben láttam, hanem egy programban. Szigethy András (PIGMY) a feltaláló. Az ő címe elég közismert, én pedig nem akarom őt magamra haragítani, ezért kérek mindenkit, hogy nála érdeklődjék. Egyébként ha a kettes generátort analóg üzemben zajra kapcsoljuk, további hangerőnövekedést érhetünk el, de tudomásom szerint ezt még senki nem rendezte be táblázatba, vagyis még senki sem használta ki a további hangerők lehetőségét. Természetesen az analóg jel használata sem ,,A MEGOLDÁS", ennek is akadnak hátrányai. A legnagyobb hátrány, hogy emellett a gépnek szinte semmire sincs ideje. Ennek az az oka, hogy a zenével itt nem annyit foglalkozunk, hogy beállítunk néhány regisztert, és a többit a hardver végzi, hanem a hangelőállítással végig foglalkoznunk kell, és egy hangjelhez legalább annyiszor kell foglalkozni egy másodperc alatt, mint Hz-ben mért frekvenciájának kétszerese, vagyis például egy 5 kHz frekvenciájú négyszögjel legalább 10000 ciklust jelent másod-

percenként (ugyanis 5000-szer „magas", 5000-szer „alacsony" állapotba kerül), egy nyolc részletben előállított szinuszjel pedig 8*5000=40000 ciklust jelent... Természetesen ilyen pontossággal időzíteni csak a TED időzítőkkel lehet, és még így is gyakran le kell kapcsolni a képernyőt. Valójában az analóg lehetőségeket legjobban digitalizált hangeffektusokkal lehet kihasználni, mint például azt Kosir Attila tette DIGITAL BALL című játékában. Ilyen minőséghez azonban speciális hardver és nagy tapasztalat szükséges. Próbáljuk meg a 3. listán látható program beütését, és kísérletezzünk a hullámformákkal (az alapbeállítás normál „A" hangot szolgáltat szinuszos jelalakkal). Nézzük meg hát digitalizátor-programunkat! Működési elve: a géphez csatlakoztatott adatmagnó a processzorport ($01-es cím) 4. bitjén küldi a gép felé a jeleket. Ezt mi állandó időközönként leolvassuk, és a bitet a memóriába írjuk. Ha a memória megtelt, befejezzük a digitalizációt. Maximum körülbelül 34000 mintavétel érhető el másodpercenként, a minimális ajánlott szám 8000 lehet. Lejátszáskor folyamatosan olvassuk a memóriát, és a kiolvasott értéktől függően alacsony vagy magas szintre állítjuk az analóg kimenetet.

Részletesebb leírás

Digitalizálás: Megkérjük a felhasználót, hogy indítsa el a magnót, és megvárjuk, míg ezt megteszi. Kereső funkcióra állunk. Megvárjuk, míg a felhasználó a SHIFT billentyű lenyomásával jelzi, hogy a digitalizálni kívánt részhez érkeztünk. Ha a magnó nem ad jelet, megvárjuk, míg ez elkezdődik, így például egy hang elé is beállhatunk a kazettával és az üres részen lenyomva a SHIFT billentyűt a gép maga keresi meg a hang kezdetét. Letiltjuk a megszakítást, majd lekapcsoljuk a képernyőt, ezáltal a gép gyorsabban és egyenletesebben fut. Beállítjuk az időzítőket. Az X-regisztert használjuk „maradék bit" számlálónak, ha itt nulla áll, a bájtban az összes bit elfogyott. A beolvasott biteket egyenként léptetjük be a pufferbe. Ha ez tele van (X = 0), kiírjuk a következő bájtba. Ezt addig folytatjuk, míg vége nincs a használható RAM-nak ($FD00-ás cím). A folyamat közben a felhasználó folyamatosan hallgathatja a jelet úgy, ahogy az majd lejátszáskor hangzani fog. Lejátszás: A megszakítást kikapcsoljuk, RAM-ra kapcsolunk, az időzítőket beállítjuk. Folyamatosan olvassuk a memóriát, és a kiolvasott bittől függően magas vagy alacsony állapotba hozzuk az analóg kimenetet. Ezt a folyamatot is $FD00-ig folytatjuk. Itt szeretném megemlíteni, hogy azonos elven működik a közkézen forgó FOONER és DIGITAL SYSTEM nevű két program is.

Köszönöm figyelmüket, a következő (a befejező) cikkben a memóriáról és a billentyűzet/joystick-kezeléséről olvashatnak, és bemutatásra kerül egy ROM-COPY is.

```
TABLA    .BYT $80,$40,$20,$10,8,4,2,1
.END

;5. resz 1. lista
;
GENLOW=$FF0F                          ;2.GENERATOR ALSO BYTE
GENHI=$FF10                           ;            FELSO
SNDVEZ=$$FF11                         ;TED HANGVEZERLO REGISZTERE
;
;
*=$2000
;
         LDX #2
         LDY #3                       ;NORMAL 'A' HANG
         STX GENLOW
         STY GENHI
         LDA #$68                     ;2. CSATORNA NEGYSZOGJEL & ZAJ
         STA SNDVEZ
         LDY #0                       ;VARAKOZO CIKLUS
         LDX #0                       ;KB. 1 MASODPERC
VAR      INX
         BNE VAR
         INY
         BNE VAR
         LDA SNDVEZ
         EOR #$20                     ;NEGYSZOGJEL KI/BE
         STA SNDVEZ
.END
;5. resz 2. lista
;
GENLOW=$FF0F                          ;2.GENERATOR ALSO BYTE
GENHI=$FF10                           ;            FELSO
SNDVEZ=$$FF11                         ;TED HANGVEZERLO REGISZTERE
TIMLOW=$04FD                          ;IDOZITO ALSO BYTE
TIMHI=$04FF                           ;        FELSO
;
;
*=$2000
;
         LDX #2
         LDY #3                       ;NORMAL 'A' HANG
         STX GENLOW
         STY GENHI
         LDA #$28                     ;2 CSATORNA NEGYSZOGJEL
         STA SNDVEZ
         LDX #0                       ;KB. 5 MASODPERC
         LDY #$FF
         STX TIMLOW                   ;IDOZITO
         STY TIMHI                    ;BEALLITASA
         BRK                          ;KILEPUNK, DE A HANG TOVABB SZOL
.END
```

```
analog hang.....page # 1

Line#  Addr Code          Source

00001  0000               ;analog hang
00002  0000                         * = $7000
00003  7000               ;
00004  7000               tempo     = $00fb        ;'a' hang periodusido/8
00005  7000               timerl    = $ff00
00006  7000               timerh    = $ff01
00007  7000               irq       = $ff09
00008  7000               ted1      = $ff06        ;ted vezerlo #1
00009  7000               ;
00010  7000               ;
00011  7000 78                      sei
00012  7001 a9 0b                   lda #$0b
00013  7003 8d 06 ff                sta ted1       ;kep ki
00014  7006 a2 fb                   ldx #<tempo    ;lejatszasi
00015  7008 a0 00                   ldy #>tempo    ;sebesseg
00016  700a 8e 00 ff                stx timerl     ;beirasa
00017  700d 8c 01 ff                sty timerh
00018  7010 a2 00         kezd      ldx #0
00019  7012 bd 29 70      beolv     lda tabla,x    ;hangerot beolvassuk
00020  7015 f0 f9                   beq kezd       ;zaro nulla?
00021  7017 a8                      tay            ;mentjuk
00022  7018 a9 08                   lda #8         ;timer1 bit
00023  701a 2c 09 ff      var       bit irq        ;megvarjuk, mig
00024  701d f0 fb                   beq var        ;lejar
00025  701f 8d 09 ff                sta irq        ;irq torlese
00026  7022 8c 11 ff                sty $ff11      ;hangerot kiirjuk
00027  7025 e8                      inx            ;mutato noveles
00028  7026 4c 12 70                jmp beolv      ;ciklushoz
            94 97 98 97
00029  7029               tabla .byt $94,$97,$98,$97 ;szinuszos
            94 91 90 91 00
00030  702d                     .byt $94,$91,$90,$91,;tablazat

end of assembly, error count = 00000

beolv   7012    irq     ff09    kezd    7010    tabla   7029
ted1    ff06    tempo   00fb    timerh  ff01    timerl  ff00
var     701a
```

```
          Line#  Addr Code        Source

          00001  0000             ;hang-digitalizator
          00002  0000             ;
          00003  0000                        * = $1001
          00004  1001             ;
          00005  1001             ;
          00006  1001             ;
          00007  1001             ide        = $d0
          00008  1001             innen      = ide
          00009  1001             puff       = ide+2
          00010  1001             timerl     = $ff00
          00011  1001             timerh     = $ff01
          00012  1001             ram        = $ff3f
          00013  1001             rom        = $ff3e
          00014  1001             bill       = $0543
          00015  1001             tedvez     = $ff06
          00016  1001             hang       = $ff11
          00017  1001             border     = $ff19
          00018  1001             port       = $01
          00019  1001             uzenet     = $e31b
          00020  1001             mtrbe      = $e33d
          00021  1001             mtrki      = $e3b0
          00022  1001             basic      = $8703
          00023  1001             tempo      = $0060
          00024  1001             irq        = $ff09
```

```
1001                 mon    = $ff52
1001                 ;
1001  0b 10 c7 07
1001                        .byt $0b,$10,$c7,$07 ;basic
      9e 34 31 30 39 00 00 00
1005                        .byt $9e,'4109',0,0,0;1991 sys4109
100d                 ;
100d                 ;
100d                 ;lejatszo
100d                 ;
100d                 ;init
100d                 ;
100d  78                    sei
100e  8d 3f ff              sta ram
1011  a2 60                 ldx #<tempo
1013  a0 00                 ldy #>tempo
1015  8e 00 ff              stx timerl
1018  8c 01 ff              sty timerh
101b  ad 06 ff              lda tedvez
101e  29 ef                 and #$ef            ;$ef-kep ki/$ff-kep be
1020  8d 06 ff              sta tedvez
1023  a0 00                 ldy #0
1025  a9 11                 lda #$11            ;$1100-tol jatszik
1027  84 d0                 sty innen
1029  85 d1                 sta innen+1
102b                 ;
102b                 ;
102b                 ;a lejatszo fociklusa
102b                 ;
102b                 ;
102b  b1 d0          ujbajt lda (innen),y       ;byte be
102d  a2 08                 ldx #8              ;bitszamlalo
102f  0a             focikl asl a               ;bit leptetes
1030  48                    pha                 ;maradek mentese
1031  b0 05                 bcs magas
1033  a9 90                 lda #$90            ;0-as hangero
1035  4c 3a 10              jmp var
1038  a9 93          magas  lda #$93            ;3-as hangero
103a  48             var    pha                 ;hangero mentes
103b  a9 08                 lda #8              ;timerl mexak
103d  2c 09 ff       varcik bit irq             ;mexakot
1040  f0 fb                 beq varcik          ;megvarjuk
1042  8d 09 ff              sta irq             ;toroljuk
1045  68                    pla                 ;hangero vissza
1046  8d 11 ff              sta hang            ;lerakjuk
1049  8d 19 ff              sta border          ;csikozunk
104c  68                    pla                 ;maradek vissza
104d  ca                    dex                 ;bitszamlalo csokkentes
104e  d0 df                 bne focikl          ;meg van
1050  c8                    iny
1051  d0 d8                 bne ujbajt
1053  e6 d1                 inc innen+1
1055  a5 d1                 lda innen+1
1057  c9 fd                 cmp #$fd            ;$fd00-ig megy
1059  d0 d0                 bne ujbajt
105b                 ;
105b                 ;
105b                 ;vege
105b                 ;
105b                 ;
105b  8d 3e ff              sta rom
105e  ad 06 ff              lda tedvez
1061  09 10                 ora #$10
1063  8d 06 ff              sta tedvez          ;kep be
1066  a9 00                 lda #0
1068  8d 11 ff              sta hang            ; hang nullazas
106b  60                    rts                 ;vege
106c                 ;
106c                 ;
106c                 ;a digitalizator
106c                 ;
106c                 ;
```

```
106c 20 1b e3          jsr uzenet      ;press play on tape
106f 20 8d e3          jsr mtrbe       ;magno motor be
1072 a5 01     hallga  lda port
1074 18                clc
1075 69 31             adc #$31        ;itt hallgatunk
1077 8d 11 ff          sta hang
107a 8d 19 ff          sta border
107d ad 43 05          lda bill        ;felhasznalo indit?
1080 f0 f0             beq hallga      ;nem
1082 a9 10             lda #$10        ;bemeno bit teszt
1084 24 01     varjel  bit port        ;1-re varunk
1086 f0 fc             beq varjel
1088 a9 0b             lda #$0b
108a 8d 06 ff          sta tedvez      ;kep ki
108d 78                sei
108e a2 60             ldx #<tempo
1090 a0 00             ldy #>tempo
1092 8e 00 ff          stx timerl
1095 8c 01 ff          sty timerh
1098 a0 00             ldy #0
109a a9 11             lda #$11        ;$1100-tol
109c 84 d0             sty ide
109e 85 d1             sta ide+1
10a0 a2 08     ciklus  ldx #8
10a2 a9 08     olvas   lda #8
10a4 2c 09 ff  vartim  bit irq
10a7 f0 fb             beq vartim
10a9 8d 09 ff          sta irq
10ac a9 10             lda #$10        ;bemeno bit teszt
10ae 24 01             bit port
10b0 d0 06             bne high
10b2 18                clc
10b3 a9 90             lda #$90
10b5 4c bb 10          jmp shift
10b8 38        high    sec
10b9 a9 93             lda #$93
10bb 26 d2     shift   rol puff        ;beshifteljuk
10bd 8d 11 ff          sta hang
10c0 ca                dex
10c1 d0 df             bne olvas
10c3 a5 d2             lda puff
10c5 91 d0             sta (ide),y
10c7 c8                iny
10c8 d0 d6             bne ciklus      ;kovetkezo byte
10ca e6 d1             inc ide+1
10cc a5 d1             lda ide+1
10ce c9 fd             cmp #$fd        ;$fd00-ig
10d0 d0 ce             bne ciklus      ;kovetkezo,byte
10d2                   ;
10d2                   ;
10d2                   ;vege
10d2                   ;
10d2                   ;
10d2 58                cli
10d3 20 b0 e3          jsr mtrki
10d6 a9 1b             lda #$1b
10d8 8d 06 ff          sta tedvez      ;kep be
10db a9 00             lda #0
10dd 8d 11 ff          sta $ff11       ;hang nullazas
10e0 4c 52 ff          jmp mon

end of assembly, error count = 00000

basic   8703   bill    0543   border  ff19   ciklus  10a0
focikl  102f   hallga  1072   hang    ff11   high    10b8
ide     00d0   innen   00d0   irq     ff09   magas   1038
mon     ff52   mtrbe   e38d   mtrki   e3b0   olvas   10a2
port    0001   puff    00d2   ram     ff3f   rom     ff3e
shift   10bb   tedvez  ff06   tempo   0060   timerh  ff01
timerl  ff00   ujbajt  102b   uzenet  e31b   var     103a
varcik  103d   varjel  1084   vartim  10a4
```

Számítógépes fogalmak

Lexikon

XII. rész

FELHASZNÁLÓI FELÜLET

Rendes körülmények között egy programot a C64-esen a LOAD utasítással töltünk be és RUN-nal indítunk. Egy felhasználói felület esetében az utasítások elmaradnak. A legjobb példa erre a GEOS. Itt egy programot nem a nevével hívunk föl, hanem egy megjelenő grafikus szimbólumra (ikonon) kell ,,rálőni", azaz a joystickkel rálépni és a tűzgombot megnyomni. Ezután a GEOS maga végzi el a program fölhívását. Vannak olyan programok is, amelyek szintén rendelkeznek felhasználói felülettel, pl. a Geowrite. Mialatt a szövegszerkesztő programok általában billentyűzet utasításokkal dolgoznak, a Geowrite menütechnikát kínál. Ennek a menünek a funkcióit joystickkel vagy egérrel és a tűzgombbal (egérgombbal) kell aktiválni. Mindent összevetve a felhasználói felületek különösen a kezdők számára könnyítik meg a kezelést. Ez nagyban csökkenti a ,,konfliktusokat" a számítógéppel.

IKON

A modern felhasználói felületeknél a programokat nem a nevek megadásával kell fölhívni. Hanem a felhasználói felületben a lemezen található minden egyes állomány egy bélyeg méretű szimbólumot kap, amelyet a képernyőn látunk viszont. A C64-es esetében a GEOS dolgozik így. A szimbólum általában valamilyen formában kapcsolódik a program nevéhez, felhasználási területéhez. Például a szövegszerkesztőt egy toll jelzi. Ha egy ilyen programot el akarunk indítani, akkor az egérrel vagy a joystickkel rá kell vezetni az ikonra a mutató nyilat, és kétszer meg kell nyomni gyorsan egymás után az egér-, (tűz)gombot. Ezzel a módszerrel nemcsak programindítás lehetséges, hanem állományok másolása, törlése, átnevezése stb. is.

ABLAKTECHNIKA

A szakzsargonban ezt windowingnek is nevezik. Ezzel a technikával manapság egyre többször találkozunk. Durván válogatva itt kommentár- és visszaigazolási ablakokkal, redőnymenükkel találkozunk. Az elsőként említettek esetében általában beviteleket kell tenni vagy visszaigazolni, esetleg hibajelzésekkel találkozunk így. A redőnymenük (Pull-Down menük) esetében a képernyőn megjelenik egy négyszög, amely úgy takarja el a képernyőn eredetileg álló információt, mintha egy ,,redőnyt" gördítettünk volna le egy felületre. Ebben a menüben azután különböző választási lehetőségeket találunk. Ha elhagyjuk a menüt, a redőny ,,fölgördül", s a képernyő eredeti állapota visszaáll. A GEOS esetében találunk a képernyőn egy menüsávot. Ha itt kiválasztjuk az egyiket (kattintás a mezőn, rálövés), automatikusan legördül a redőnymenü. A megjelenő választási lehetőségek közül valamelyiket aktiválva végrehajtjuk a kívánt műveletet. Az ablaktechnika megkönnyíti a számítógéppel a munkát, hisz a parancsok bonyolult bevitele elmarad.

PANEL

Más szóval NYÁK, azaz nyomtatott áramkör. Az elektronikában az alkatrészek hordozója legtöbbször egy üvegszállal megerősített műgyanta lap, ahol a drótok helyett az összekötéseket maratott vagy felnyomtatott vezetékpályák látják el. Mivel a panelok előállításakor sablonokkal dolgoznak, ezért a hibás összeköttések lehetősége majdnem lehetetlen. A lapokat leggyakrabban valamilyen norma szerinti méretben készítik, és azon furatokat is találunk az építőelemek számára, amelyeket forrasztással össze lehet kötni a vezetékpályákkal. Mivel a panel rendelkezik egy úgynevezett érintkezősávval, ezért azt gyakran modulkártyának is nevezik, mivel ezeket egy előre elkészített csatlakozóba (slot) egy mozdulattal be lehet helyezni. A gyártásnál először ,,layout"-okat készítenek, amelyeket a panel fényérzékeny felületére ultraibolya fénnyel ráfotóznak. A fölvitt réteg nem megvilágított részét a panelról le lehet maratni, így azon csak a layoutnak megfelelő vezetékpályák maradnak.

KÉPERNYŐMASZK

Hogy a felhasználók számára a programokban az adatok bevitelét megkönnyítsék, ezért a képernyőn egyfajta formulárt építenek föl. A rendelkezésre álló felületet magyarázó szövegekkel ellátott részekre osztják, ahova azután a kívánt adatokat be lehet vinni. Ha a maszkokat áttekinthetően készítették el, úgy az nagyban hozzájárul a hibamentes adatbevitelhez. Ezen kívül például a jól elkészített maszkok beviteli mezői lehetővé teszik az inputok ellenőrzését is, például a hosszúságra. Természetesen azt is ellenőrizni lehet, hogy a bevitt adat szám-e vagy karakterlánc (sztring). Ezekre azután a jó program reagálni tud. Ha valaki dolgozott már a Superbase programmal, akkor tudhatja, hogy a képernyős (beviteli) maszkokat nem csupán a szép megjelenítés miatt találták fel.

Még többet észszel!

IX. rész

Az első kérdésemre, a kettős IF (1. pl.) helyettesítésére elég könnyű a válasz, a megoldás a 2. példa. A két változat elvileg látszólag tökéletesen egyenértékű mindkét esetben a PRINT akkor és csak akkor hajtódik végre, ha A és a B is igaz (nem nulla). Most mégis a két változat különbségeiről fogok beszélni.

Nem tudom, ki hogy van vele, nekem az AND-es változat szimpatikusabb, talán mert áttekinthetőbb. Sajnos úgy tűnik, ez a legtöbb jó, ami elmondható róla. Egyik gyengéje a viszonylagos lassúsága. A kettős IF-ben a B vizsgálatához csak akkor jutunk, ha a feltétel A tagja igaz, az AND művelet viszont minden esetben elolvassa és kiértékeli mindkét tagot. Ráadásul az AND végrehajtása, úgy látszik, lényegesen lassabb az IF-nél, bizonyára a többszöri adatkonverzió miatt. Még azzal sem tudtam a próbák során az IF fölé kerekedni, hogy egész típusú változókat használtam feltételnek.

A másik különbség az, hogy vannak esetek, amikor az IF-ek simán lefutnak, de az AND hibával leáll. Például K = —1 értékkel a 3. példa elkerüli a buktatót, azaz a tömb negatív indexű elemének lekérdezését, a 4. példa meg nem. Itt hiába adom meg szűrőfeltételként „K > = 0" tagot, és az hiába nem teljesül, az AND kiértékeléséhez a gép azért még megnézné a tömböt is, ami persze nem lehetséges.

Van egy másik lehetőség a hibára: A = 2 B = 4 értékkel próbáld ki az 1—2. példákat! Ha nem világos, hogy az AND-es változat miért nem működik, akkor jó lenne még egyszer átfutnod a VI. részt, ahol a logikai műveletek finomságait tárgyaltuk. A 2 AND 4 ugyanis egy művelet, amelynek az eredménye 0, azaz „hamis".

Tanulság: az AND olyan, mint a nő. Szerethetjük nyugodtan a szépségéért, ha sosem feledjük a csalfaságát.

És most csapjunk bele a másik témába! Remélhetőleg ez is szolgál majd hasznos ismeretekkel. Kiválasztottam egy olyan kis programot, amely az újságban jelent meg (92/5., 8. oldal, keresd meg, kérlek), és amelyet én másképp írtam volna meg, összhangban az eddig bemutatott elveimmel. Miközben elmondom a saját ötleteimet, felidézhetjük kicsit a tanultakat, és felvethetek újabb témákat is. Ezekkel a kitérőkkel biztosan sokáig ki fog tartani ez az egy példa, de nem fogunk unatkozni.

```
(1.)
20 IF A THEN IF B THEN PRINT "*"

(2.)
20 IF A AND B THEN PRINT "*"

(3.)
20 IF K>=0 THEN IF D(K) THEN END

(4.)
20 IF K>=0 AND D(K) THEN END

(5.)
40 REM *** KURZOR+PRINT * (C64)
41 REM * SYS W,SOR,OSZLOP [,NYOMTLISTA [;]]
42 W=49152: FOR I=0 TO 33: READ V: S=S+V: POKE W+I,V: NEXT I
43 READ V: IF S<>V THEN PRINT "HIBA": END
44 DATA 32,253,174,32,158,183,138,72,32,253,174,32,158,183,138,168,104
45 DATA 170,24,32,240,255,32,121,0,240,6,32,253,174,32,160,170,96,4321

(6.)
1 XV$=CHR$(19): FOR I=1 TO 25: XV$=XV$+CHR$(17): NEXT I: REM HOME & LE
2 XH$="": FOR I=1 TO 39: XH$=XH$+CHR$(29): NEXT I: REM JOBBRA
10 XV=3: XH=15: GOSUB 99: PRINT "4.SOR 16.OSZLOP"
11 ...
98 REM ====== KURZOR (XV,XH;)
99 PRINT LEFT$(XV$,XV+1);LEFT$(XH$,XH);: RETURN

(7.)
190 N$="": FOR N=5 TO 32: GET A$: N$=N$+A$: NEXT N
200 FOR N=23 TO 20 STEP -1: IF MID$(N$,N,3)<>"SEQ" THEN NEXT N:
      GOTO 250
210-230
240 N$(Z)=MID$(N$,N-19,18): Z=Z+1
```

Először szólnom kell a program (egyetlen) hibájáról. A 370. sorban látható SYS W arra lenne hivatott, hogy a kurzort a megadott pozícióra (felső sor 22. oszlop) vezényelve kiírja a lemezegységtől kapott hibaüzenetet. Mivel a C64 BASIC-je (a program ebben íródott) nem segít bennünket a kurzor közvetlen áthelyezésében, a szerző egy kis, e célra készült gépi kódú programmal oldotta meg ezt a feladatot. Ez nagyon gyakori és egyszerű mód a BASIC-ben nehézkesen kezelhető problémák áthidalására. A SYS utasítás a megadott címen kezdődő (és korábban oda betöltött) gépi kódú programot szubrutinként hívja meg, azaz amikor az befejeződött, a BASIC program a következő utasításon folytatódik. Rendes esetben.

Nem rendes eset például, ha a program kezdőcímét pontatlanul közöljük a SYS utasításban a géppel, amely sajnos roppant kényes az ilyen apróságok gondos kezelésére. A példaprogramban a kezdőcím a W változó aktuális értéke. Mivel azonban ez az egyetlen hely, ahol a W a programban szerepel, természetesen ez az érték nulla. Az viszont biztos, hogy ott nincs a kívánt kis szubrutin, mert ott más, a gép számára rendkívül fontos byte-ok foglalnak a tárban helyet. Ilyen állapotban futtatva a programot még csak azt sem lehet megmondani, hogy a SYS utasításnak mi lesz a hatása (ez a tár pillanatnyi tartalmától és a processzor típusától függ), de arra mérget vehetünk, hogy semmi jó. A következő dolgok maradtak el a programból: a kurzor kezeléséhez szükséges gépi kódú szubrutin közlése, a programban vagy a leírásban utalás ennek betöltésére, és a W = kezdőcím utasítás elhelyezése a program elején. Szerencse, hogy ez a kínos hiányosság csak lemezhiba esetén okoz galibát, ha ez vígasztal valakit.

Azért, hogy továbbmehessünk, az 5. példában megadom a hiányzó részt, BASIC feltöltős alakban. A mi esetünkben elég hozzáírni ezeket a sorokat a programhoz, de önállóan is felhasználhatók. A bővítést úgy írtam meg, hogy W-nek bármilyen értéket adva működőképes legyen. A SYS után meg kell adni a sor és oszlop értékeit, és utána a PRINT-hez hasonlóan megadhatók a kiírandók is. Ahogy az a 370. sor tanúsága szerint eredetileg lehetett.

Van, amikor nem alkalmas a helyzet a gépi kódú szubritonok használatára, vagy éppen nem akarjuk a BASIC területét elhagyni (mint például a nemrég befejeződött játékpályázatban). Ilyenkorra van más megoldás is. A ,,POKE 214,sor: POKE 211,oszlop: SYS 58732'' utasítással a kurzort a kívánt helyre küldhetjük. A művelet lényege, hogy miután feltöltjük a megfelelő értékekkel azokat a rekeszeket, amelyekben a gép a kurzor pozícióját tárolja, meghívunk egy szubrutint, amely a kurzort a tárolt értékek szerinti helyre viszi. Sajnos ez csak a C64-esen próbálható ki, de biztos vagyok benne, hogy más gépeken is lehet hasonló trükköt alkalmazni.

Úgy tisztességes, ha megemlítek egy közismert megoldást, amely minden Commodore gépen megvalósítható. Ebben kihasználjuk, hogy a kurzorvezérlő jelek karakterláncban tárolhatók, és kiíráskor végrehajtódnak (6. pl.). Csak emlékeztetőül: a bal felső sarok koordinátái hagyományosan 0;0.

Lépjünk tovább! Felmerülhet a kérdés, hogy miért kellett lényegében az egész programot egy szubrutinba tenni, amiért is a ,,főprogram'' csupán három rövid sor. Általában ennek valóban nincs sok értelme, az olvashatóságot ez nem nagyon javítja. Ez esetben viszont a szerző egy olyan programot mutatott be, amelyet véleménye szerint a saját programunkban egy szubrutinként fogunk elhelyezni, ha alkalmazzuk. A ki- és belépéseket tehát ennek megfelelően írta meg, nekünk a megfelelő átsorszámozás az összes feladatunk.

Feltűnhet, hogy a nulla helyett, összesen hat esetben, csak egy tizedespont látható. Az az igazság, hogy az ,,A = 0'' határozottan lassabb, mint az ,,A = .'' utasítás. Ennek az oka is biztosan a különböző számábrázolási alakok közötti átalakítások és az interpreter olvasási algoritmusa környékén keresendő. Jó, ha ez is eszünkbe jut, ha egy programot próbálunk felgyorsítani.

Az a kár, hogy egy program olvasása közben egy pont lassabban juttatja eszembe a nullát, mint egy nulla, és szerintem ezzel sokan vannak még így. Itt megint ütközni látszik a megszokás, az ízlés a hatékonysággal. Persze fontosnak tartom a gyorsaságot, sokszor utaltam is rá, de csak az ésszerűség határáig. A hat eset közül csak egyetlen olyan van (190. sor), ahol a nulla minden filenév elolvasásakor sorrakerül, az összes többi nullás értékadás az egész programban csak egyszer (!) hajtódik végre. Nyilván belátható, hogy mennyire káros az olvashatóságot feláldozni egy olyan gyorsításért, amellyel nagyjából talán egy ezredmásodpercet javíthatunk egy másodperces nagyságrendű művelet idején, nem beszélve akkor az egész program sokmásodperces futásáról.

Szerintem minden gyorsítást érdemes előzőleg külön kipróbálni, ha nem vagyunk egészen biztosak a hatásában. És csak akkor van értelme bevetni, ha a nyereség megéri a változtatásba befektetett munkát, és nem veszítünk el vele valami értékesebbet. Valószínűleg nem igazolható szabályosan, de tapasztalatom szerint mégis igaz, hogy egy (a programozó képességei szerint) elég jól megírt program futási ideje és az általa felhasznált tár mértéke fordított arányban áll egymással. Ez például azt jelenti, hogy ez a programozó számottevő gyorsulást csak tárveszteség árán érhet el. Az optimum megtalálása időrabló feladat, de szerencsére ritkán kell vele foglalkozni.

A kép teljességéért mutatok néhány valóban hatékony fogást a gyorsításra. Kezdhetjük azzal, hogy a tömbváltozók használata mindig lassabb a skalárokénál. Az olvasáskor tehát ne rögtön a tömbbe, hanem egy N$ nevű átmeneti változóba gyűjtsük az adatokat. És csak akkor másoljuk azt a tömbbe, ha azt valóban meg akarjuk tartani, vagyis ha SEQ típusú. Ez utóbbitól nem lesz gyorsabb a program, csak értelmesebb. Egyébként meg kell jegyeznem, hogy ha csak kiíratni szándékozunk a fileneveket, akkor felesleges őket előtte egy tömbben tárolni, ha viszont felhasználásra összegyűjteni akarjuk, akkor értelmetlen a tömb törlése a 280. sorban.

Sajnos most nem mondhatom el, hogy a ,,$'' nevű file részletesen mit tartalmaz, de annyit tudni kell, hogy 32-byte-os adagokban kapjuk a fejlécet, a bejegyzéseket (neveket) és a szabad blokkok kiírását, de az első négy karaktert mindig lehagyjuk (180, 250). A típus jele az N$ 23—20. pozíciójától kezdődik, attól függően, hogy a file hossza hány jegyű szám (1—4). Számíthatunk a rövidebb file-okra, ezért kezdjük a típus keresését a 23. helytől (200). És végül kihasználhatjuk, hogy a név és a típus együtt mozog a hossztól függően, ezért a név előtti idézőjel mindig 19 karakterrel a típus előtt lesz (240). Ezzel megtakarítunk egy keresést, és egyben felkészítettük a programot a 0—9999 blokkig terjedő hossz kezelésére, vagyis megszüntettünk egy újabb hiányosságot is. A javításokat a 7. példán mutatom be, a beolvasási idő rövidülése kb. 30 százalék. Az, hogy közben nem vesztettem tárat, nekem azt jelzi, hogy a program még nem volt elég jó.

Legközelebb ismét erre a programra alapozom a mondandómat, ehhez előkészítésként van három kérdésem.

A 180. és 250. sorban négy-négy karaktert olvasunk be úgy, hogy azokat nem használjuk semmire. Miért jó, hogy ez kétszer szerepel? Mi lenne, ha inkább a 190. sor ciklusában végeznénk el ezt is?

Egy file-ból olvasni ,,GET #'' utasítással kell, és nem sima ,,GET''-tel. A program mégis működik, de vajon miért?

Főleg C64-tulajdonosokhoz: mit csinál a SYS utasítás a 180. és 250. sorokban?

Hódi Gyula

Hobbi **ELEKTRONIKA**

C-64 bővítések

EPROM Bank C64-hez

Ha Ön a C64-et nem csak játékra, hanem a mindennapi munkája során hasznos segítőtársként — például adattárolásra, méréstechnikai feladatok elvégzésére, könyvelésre, szövegszerkesztésre — alkalmazza, bizonyára érzi a kellemetlenségét a sok unalmas billentyűzésnek, programkezelésnek, lemezcserélésnek.

A most ismertetésre kerülő EPROM Bank kiküszöböli ezeket a problémákat és egy panelre sűrítve az eddigi cartridge-ek előnyös tulajdonságait, 250 kilobájtnyi kapacitásával nagyon hasznos segítőtársunk lehet. A Bank lehetőséget nyújt a felhasználónak arra, hogy a kedvenc programjait a rendelkezésre álló ROM területen elhelyezze, innen egy gombnyomásra 1—2 másodperc alatt beolvassa. Ezek a programok lehetnek lemezkezelők, gyorsítók, másolók, szövegszerkesztők stb. A tárolandó programokat a felhasználó választhatja ki, a Bank szervezőprogramja segítségével pedig egy lemezre megszerkesztheti az EPROM-ok tartalmát. Az így megszerkesztett fájl egy EPROM-égető segítségével az EPROM-ba bemásolható.

Lehet, hogy ez az első olvasásra bonyolultnak tűnik. Megnyugtatok minden kedves olvasót, hogy nem kell megijedni: aki tudja, hogy a C64-et hol kell bekapcsolni, nyugodtan hozzáfoghat a műveletek elvégzéséhez.

Az EPROM Bank előnyei

— A rendelkezésre álló tárolóhely a lemez kapacitásának másfélszerese.

— A számítógéphez való csatlakoztatás nagyon egyszerű.

— Tetszőleges nyelven írt programok tárolhatók.

— A programok gombnyomásra indulnak.

— A szoftver menüvezérelt technikát alkalmaz.

— A tartalomjegyzék ABC sorrendű.

— A ,,→" karakter a kiválasztott programra mutat.

A működtető szoftver

A Bank EPROM-jainak szervezését, feltöltését, kiválasztását, beolvasását egy 2764-es tokban elhelyezett vezérlő szoftver végzi. Ez a hardver része és nincs beszámítva azon memóriák közé, amelyek a programokat tárolják.

A szoftver menüvel vezérelhető. Utasításait a gép egy sorban közli. Tételezzük fel, hogy a felhasználó EPROM Bankja tele van programmal. A Bank a program behívására kétféle lehetőséget nyújt. Figyelembe veszi, hogy korábban már használtunk valamilyen autostartos programot és szeretnénk ezt változtatás nélkül használni a Bank-ban. Ilyen program felépítésével már foglalkoztam a Hobby sorozatban, konkrétan a 92/10-es számban. Ott PLOFI néven ismertettem az autostartos kártya működését, ezért erre részletesen most nem térek ki. Lényege: az EPROM Bank szervező programja — a továbbiakban röviden Bankmester — minden EPROM-helyhez hozzárendel egy-egy funkcióbillentyűt, összesen nyolcat. Ezek valamelyikét működtetve újabb menü jelenik meg és kigyullad a hozzátartozó LED, amely később emlékeztet arra, hogy a nyolc közül melyiket választottuk. A menü segítségével választhatunk EPROM-típust az alábbi feltételeket is figyelembe véve.

F1 = 2764; 8 Kbyte-os. Működtetése után az autostartos program indul.

F3 = 27128; 16 Kbyte-os. Működtetése után újabb almenü jelenik meg:
- F1 = alsó 8K
- F3 = felső 8K

F5 = 27256; 32 Kbyte-os. Működtetése után újabb almenü jelenik meg:
- F1 = első negyed 8K
- F3 = második negyed 8K
- F5 = harmadik negyed 8K
- F7 = negyedik negyed 8K

Láthatjuk, hogy ez a típusválasztási rendszer nem túl hatékony, mert sokáig tart a programkeresés, hiszen ha a gép adta lehetőséget tekintjük, 32 programot tudunk a Bankban elhelyezni. Ide kívánkozik még, hogy legtöbb használható program nem fér el 8 kilobájton, ami külön problémát okoz. A 8 kilobájtos szeleteket össze lehetne fűzni egy kis gépi kódú rutinnal, de ehhez komoly gép- és szoftverismeret szükséges. A fentiek miatt nem ajánlatos a programkeresésnek ezt a módját követni, mert ennél sokkal jobb és egyszerűbb a következő pontban ismertetésre kerülő SPACE-billentyűvel történő programkeresés.

Ha a SPACE (szóköz) billentyűt lenyomjuk, akkor a képernyőn megjelenik a Bankban tárolt programok címlistája. A címek ABC-sorrendben követik egymást, nem pedig az EPROM-ban elfoglalt helyük sorrendjében. A programnév előtt egy ,,→" jel látható, amely a kurzormozgató gombbokkal léptethető. Ha a nyíl a kiválasztott programra mutat, a RETURN gombbal aktivizálhatjuk. A főmenübe való visszatérés az INS/DEL billentyűvel lehetséges. A ,,↑" gomb megnyomáával a Bankmestert aktivizálhatjuk. A Bankmester lehetőséget nyújt arra, hogy a Bank tárolóit a program szükségletének megfelelően töltse fel. A számításba jöhető EPROM-ok és a tárolandó programok hosszának függvényében többféle megoldás kínálkozik.

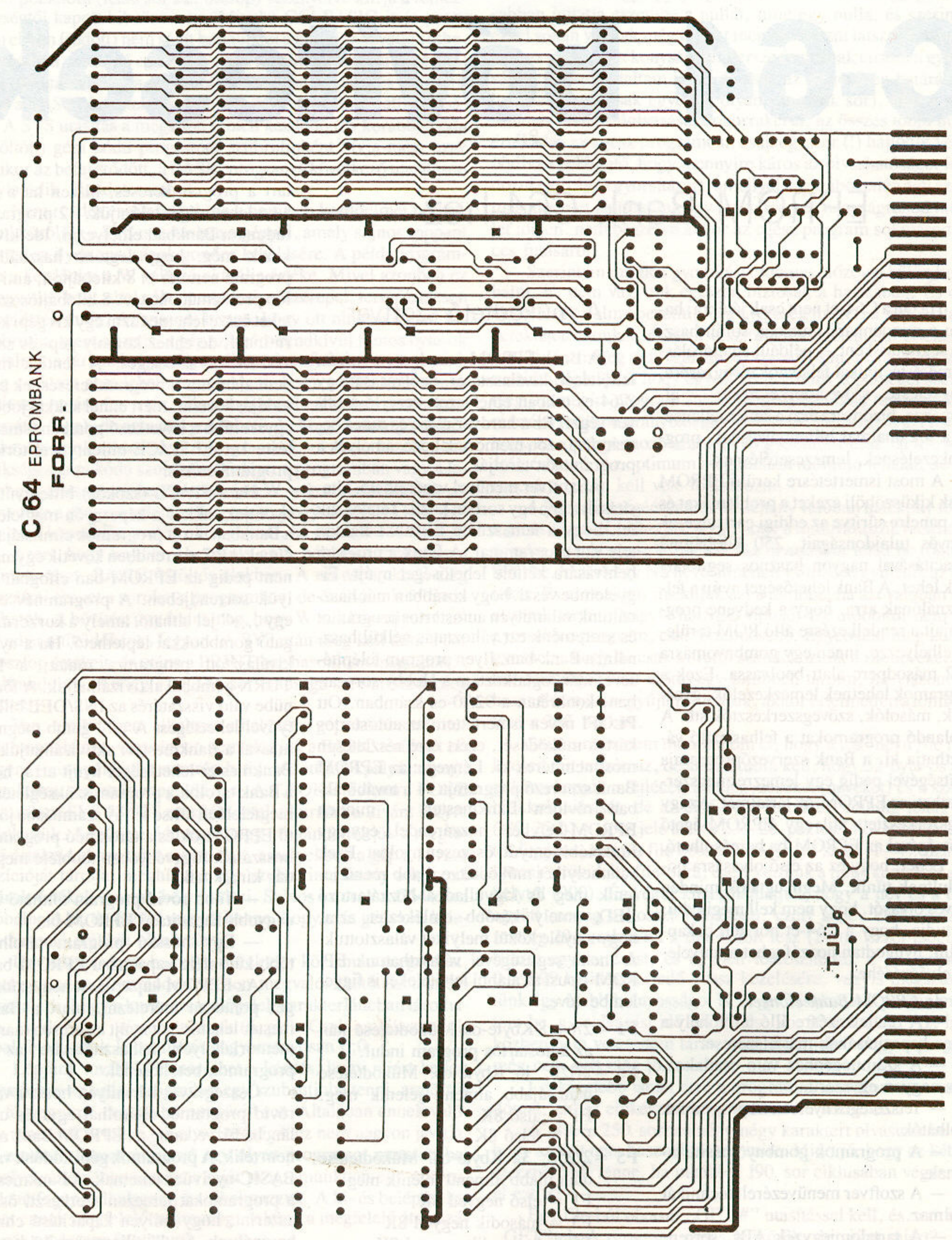
— Több rövid program tölthető egy nagyobb kapacitású EPROM-ba.

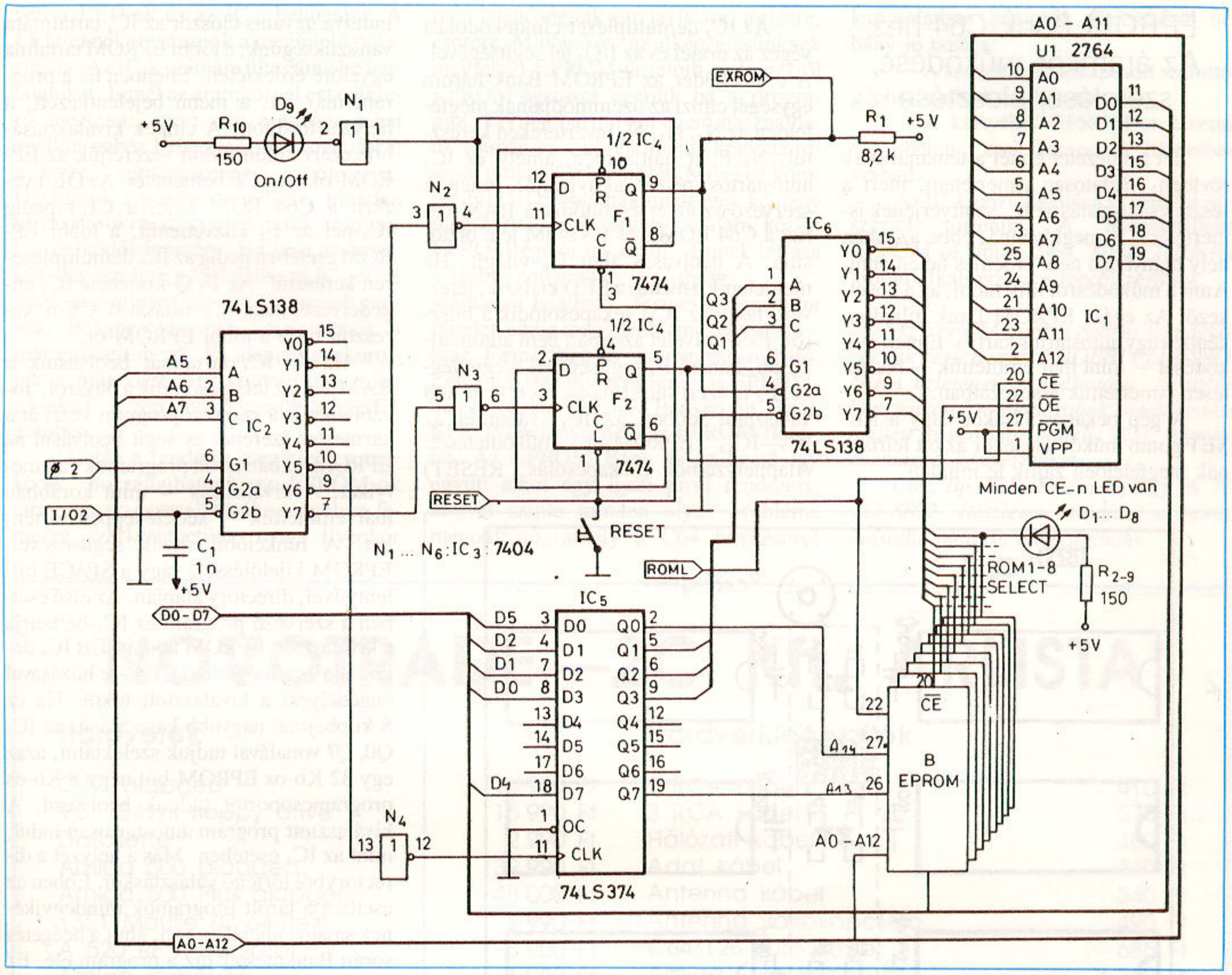
— Egy hosszú program tárolható több különféle kapacitású EPROM-ban.

Az EPROM kapacitását nem szükséges pontosan méretezni, mert a Bankmester lehetőséget nyújt az üresen maradt memóriahelyek felhasználására az új programok betöltésénél.

Összegezve: bármilyen hosszú vagy rövid programot eltárolhat egymás után láncba fűzve, amíg az EPROM Bank meg nem telik. A programok gépi kódúak vagy BASIC-nyelvűek lehetnek. A Bankmester a programokat lemezen szerkeszti össze aszerint, hogy milyen kapacitású chipet használunk. A ,,↑" billentyű működtetése után az alábbi menüből választhatunk:

F1 = DIRECTORY: a lemez tartalmát listázza


C64 EPROMBANK
FORR.



1. ábra

F3 = BANKMESTER: további almenüt hív
F8 = FŐMENÜ: vissza a főmenübe.

Ha *F3*-at hívjuk, a gép megkérdezi, hogy mit kell betölteni. Válaszként gépeljük be pontosan a tárolandó program nevét. Ha nem emlékeznénk a névre: RETURN majd *F1*-gyel írassuk ki a könyvtárat. A programnév pontos beírása után a következő felirat jelenik meg: „PROGRAM NEVE AZ EPROM-ban?" Ha a néven változtatni szeretnénk, gépeljük be az új nevet (max. 16 karakter), ha változatlan marad: RETURN. Ez azért lényeges, mert a kész EPROM programja a Bank katalógusában a megadott néven fog bejelentkezni. A név begépelése után változik a menü:

HOGYAN INDUL A PROGRAM?
F1 START = RUN
F3 START = SYS

A válasz: BASIC programoknál *F1*, gépi kódú programoknál *F3*. Bármelyiket választjuk, a menü következő kérdése:

MILYEN EPROM-OT VÁLASZT?
F1 ÚJ EPROM — 2764, 27 128, 27 256)
F3 RÉGI EPROM — Ez már tartalmaz a Bank-ban futó programot, de még van benne üres hely.
F5 ÚJRA IND. — Vissza a menü elejére.

F1 választása esetén a menü további kérdése egyértelmű.

F3-nál más a helyzet. A menü kéri a régi adatokat tartalmazó lemezt. Ez az a lemez, amelyről az EPROM jelenlegi tartalmát beégettük. A lemezen a program „/1" karakterekkel kezdődik. Ez a két karakter jelzi, hogy a program elő van készítve a Bank számára. Ezt az új szoftverhez is hozzáfűzi a Bankmester. Ha bemásoltuk, kéri az új lemezt. A menü további pontja utasítást ad az adatmentésre. Mentés után a directory a két összefűzött program közül csak az új nevet tünteti fel.

Ha az új program nagyobb memóriaterületet igényel, mint amennyi a „régi" EPROM-ban még rendelkezésre állt, a program ezt a képernyőre kiírja és új chip behelyezését kéri a fájl fennmaradó részének beégetéséhez. A Bankmester által szerkesztett programokat célszerű üres lemezre menteni. Ezeket később EPROM-égetővel lehet a memória chipbe beírni. Célszerű a C-újság 93/1 számában, a Hobbi sorozatban ismertetett EPROM-égetőt használni, mert annak szoftvere hatékonyan támogatja a lemezkezelést. A legjobb megoldás az, ha az égető szoftverét a Bank egyik EPROM-ja tartalmazza, ezzel egy hatékony, rugalmas rendszerhez jutunk, ami jelentősen meggyorsítja a munkát. Ha égetővel már rendelkezünk, be kell szereznünk egy EPROM-törlőt is, hiszen előfordul, hogy egy-egy Bank-ban tárolt program elavul, mást szeretnénk helyette futtatni. A régi tartalom törlése után az új program az eddigiek ismeretében egyszerűen beírható az EPROM-ba. EPROM-törlésre jól használható a közönséges kvarclámpa vagy az üvegburájától óvatosan megszabadított higanygőz lámpa is.

A leírt funkciókat megvalósító áramkör kapcsolási rajza látható az *1. ábrán*.

EPROM Bank C64-hez
Az áramkör működése, szerelése, élesztése

Ezt a fejezetet ennél a témánál csak röviden, vázlatosan ismertetem, mert a részletes tárgyalás az IC, szoftverjének ismertetését is megkívánná, ebbe azonban hely hiányában nem érdemes belemenni. Amit a működésről illik tudni, az a következő. Az egész EPROM Bank tulajdonképpen egy autostartos kártya. Ennek működését — mint már említettük — részletesen ismertettük a sorozatban.

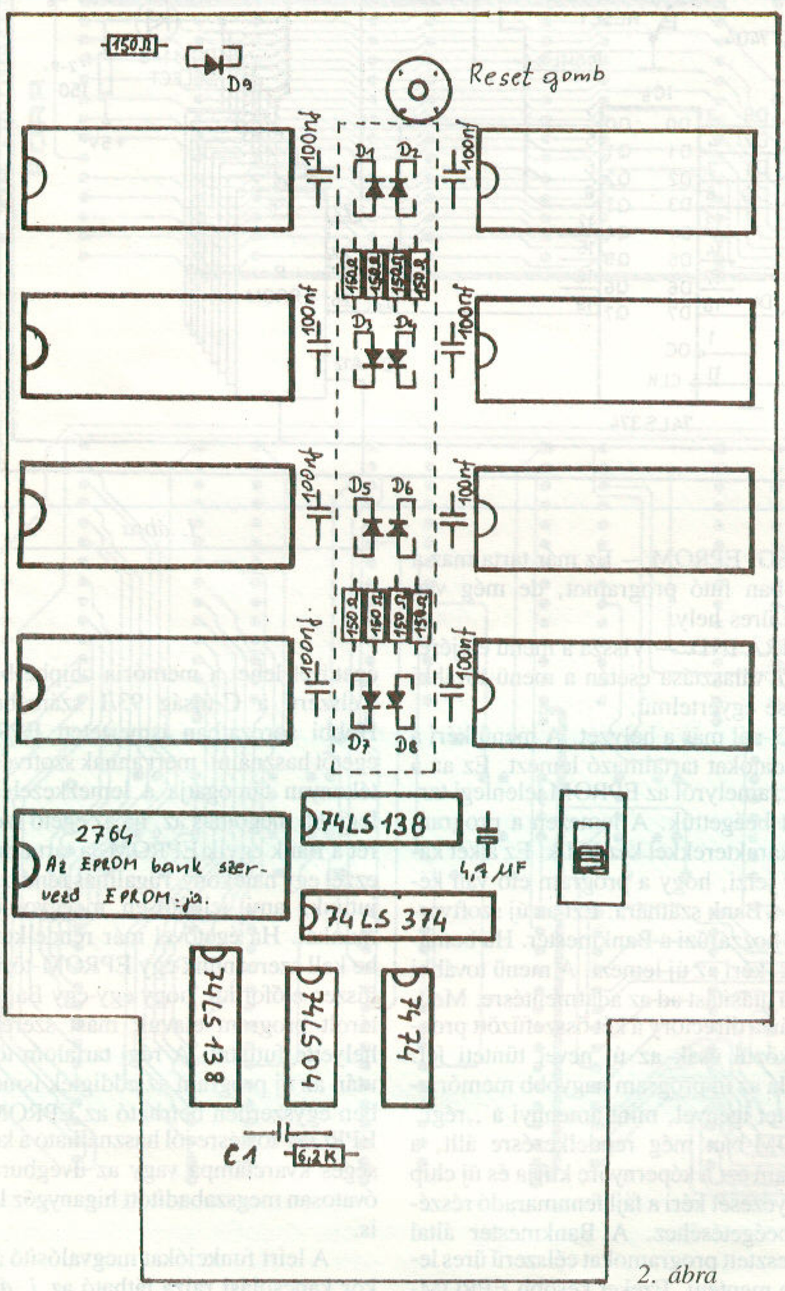
A gép bekapcsolásakor vagy a RESET gomb működtetésekor az ott leírtaknak megfelelően zajlik le minden.

Az IC_2 demultiplexer címdekódolást végez az órajel és az I/Q_2 jel segítségével. A címdekóder az EPROM Bank három egységét címzi az üzemmódjainak megfelelően az N_4, N_2, N_3 invertereken keresztül. N_2 F_1-et hajtja meg, amely az IC_1 autostartos részét aktivizálja. Innen a szervező szoftver betöltődik a RAM-ba. Ezt a C64 ROML és EXROM jele biztosítja. A beolvasás alatt D_9 világít. Ha megjelenik a menü, a LED kialszik, jelezvén, hogy az IC_3l lekapcsolódik a buszról. Ez a művelet azonban nem automatikusan, hanem F_2 segítségével megy végbe. F_2-re azért van szükség, mert ketté kell választani időben az IC_1, valamint az IC_7—IC_{13} EPROM-ok működtetését. Alaphelyzetből (bekapcsolás, RESET) indulva ugyanis először az IC_1 tartalmára van szükségünk, a többi EPROM tartalma egyelőre érdektelen. Ellenben ha a program már fut, a menü bejelentkezett, a helyzet fordított. A chipek kiválasztásához ezért külön-külön vezéreljük az EPROM-ok OE, CE bemenetét. Az OE-t vezérli a C64 ROM L-je, a CE-t pedig IC_1-nél az F_2 közvetlenül, a többi EPROM esetében pedig az IC_6 demultiplexeren keresztül. Az F_2 Q-kimenete IC_1 engedélyezéséhez IC_6 tiltásával CE-n keresztül tiltja a többi EPROM-ot.

Ha az IC_1 tartalmát beolvastuk a RAM-ba, az lekapcsolódik a buszról. Továbbiakban a szervezőprogram veszi át a karmester szerepét és segít beolvasni az EPROM-okban tárolt programok valamelyikét. A kiválasztás — mint korábban már említettük — kétféleképpen történhet. A funkcióbillentyűk segítségével, EPROM kijelöléssel, vagy a SPACE billentyűvel, directory alapján. Az első esetben a szervező program az IC_5-be beírja a kiválasztott EPROM kódját. Ezt IC_6 dekódolja és a megfelelő CE L-re húzásával engedélyezi a kiválasztott tokot. Ha ez 8 kilobájtnál nagyobb kapacitású, az IC_5 Q0, Q7 vonalával tudjuk szelektálni, azaz egy 32 Kb-os EPROM-ból négy 8 Kb-os programcsoportot tudunk beolvasni. A kiválasztott program autostartosan indul, mint az IC_1 esetében. Más a helyzet a directoryból történő választáskor. Ebben az esetben a tárolt programok mindegyikének sajátos iniciáléja van, amit a beégetés során Bankmester tűz a program elé. Ez teszi lehetővé, hogy egy hosszabb programot beolvashassunk akkor is, ha az több EPROM-ban van eltárolva. Szólni kell még a C_1 kondenzátor szerepéről, mert a gyakorlati tapasztalatok szerint nem minden C64 esetében van rá szükség. Sorozatszámokat itt nem említek. Eddig úgy tűnik, hogy a régi dobozolású gépeknél nem kell, az újaknál szükséges, mert enélkül nem jelentkezik be az EPROM Bank. A kondenzátor az I/O_2 jelet késlelteti.

Az áramkör csak jó minőségű furatgalvanizált panelre építhető meg. Mivel az a sok IC miatt finom rajzolatú, csak megfelelő gyakorlattal és lehetőséggel rendelkezők tudják elkészíteni.

Egyszerűbb a kész furatgalvanizált nyák-ot a szerzőnél megvásárolni. Címe, árajánlata a hirdetésben megtalálható. A nyák mérete 105×170 mm. A panelt az ültetési rajz formátumának megfelelő méretre kell reszelni úgy, hogy a bővítőport csatlakozójához kényelmesen illeszkedjen. Ezután fordítsuk lefelé a panel „FORR." jelzésű oldalát. Az ültetési rajz (*2. ábra*) szerint tegyük helyükre az ellenállásokat és forrasszuk be őket. Ezt kö-



2. ábra

veti a LED-ek és az IC_{12} beültetése. A LED-eknél vigyázzunk a megfelelő polaritásra. Az IC-k számára ültessünk be foglalatokat. Ennél az áramkörnél ezt célszerű megtenni, mert egy esetleges hibát enélkül nehéz behatárolni. A foglaltokat a szerző egységcsomagja tartalmazza. Ügyeljünk a gondos szerelésre, hibátlan forrasztásokra, mert ezen múlik az áramkör működőképessége. Ha kész az áramkör, dugaszoljuk a foglalatba az IC_1-et. A Bank-ot csatlakoztassuk a kikapcsolt géphez. Bekapcsolás után bejelentkezik a már említett menü. Ez alapján gyakoroljuk a működést. Ha szoftverhiba miatt „kiakad" a gép, működtessük a RESET gombot.

Miután a rendszer megfelelően működik, hozzáfoghatunk az EPROM-ok tartalmának összeállításához, melyet lemezre szerkeszthetünk meg. Ilyenkor még lehetőségünk van javításra, törlésre, cserére. Miután kész az elképzelésünknek megfelelő EPROM-tartalom, a lemezről EPROM-égetővel égetjük be a programot. Elvileg bármilyen égető használható, csupán az a követelmény, hogy az olvasni tudja a C64 lemezét. Mivel ez kompatibilitási problémák miatt igen kemény megszorítás, célszerű az általam közölt EPROM-égetőt használni. Ennek leírása a Hobbi sorozatban, a C-újság számában található. Természetesen mint minden eddig megjelent téma, egységcsomag formájában az is megvásárolható. Létezik hozzá olyan program is EPROM-ba égetve, amely csak a Bankba fut. Ez az EPROM behelyezhető az EPROM Bankba. Az EPROM Bank és az égető így együtt alkot egy igazi profi rendszert, amivel szinte minden olyan probléma megoldható, amely a C64 bővítésével kapcsolatban felmerül. Befejezésül néhány jó tanács:

— A kártyát csak kikapcsolt számítógéphez szabad csatlakoztatni.

— A kártyából EPROM-ot kivenni vagy betenni csak kikapcsolt állapotban szabad.

Az IC-k pozicionálása az ültetési rajz szerint történik. A felhasználható EPROM-ok típusa 2764, 27128, 27256 (ez utóbbi alkalmazása a legcélszerűbb).

Az EPROM Bankot építeni szándékozóknak segít a szerző HOBBI ELEKTRONIKA üzletének szolgáltatása. Az üzletben működő mintadarabok megtekintése mellett vásárolhat panelt, IC-t, egységcsomagot, készterméket.

Cím: Bp. VII., Dózsa György 16. T.: 122-8892. Részletes árajánlat a borító második oldalán megtalálható.

MORSE-gyakorló

A C-Újság régebbi számait böngészve, az 1991/4. számban találkoztam egy morse-tanuló (tanító?) programmal. Sajnos a program didaktikája eleve téves. A morse ábc-t ugyanis ,,füllel" kell megtanulni anélkül, hogy az egyes betűk jelcsoportját ,,pont-vonás", kombinációként jegyezzük meg. Így tehát pl. az ,,a" betű nem ,,.–", hanem ,,ti/tá". (Én még ismertem olyan vasúti távírászokat és forgalmistákat, akik félálomban is felismerték a távírógép kopogásából a hívójelüket, és füllel vették a szöveget.)

Előkerestem hát a néhány éve pihenő morse-gyakorló programomat, amely némi átdolgozás után a lista szerinti formát öltötte. Három üzemmódban dolgozik:

1. A beírt betűk megjelennek a képernyőn, majd megszólalnak.
2. A képernyőre írott szöveg ,,néma", de lemezre menthető.
3. A lemezről behívott szöveg folyamatosan kiíródik a képernyőre. Minden betű után megszólal a morse jele.

,,Hangzó" üzemmód választása esetén beállítható a hangmagasság és a jelsebesség.

A hangmagasság beállításakor egy, csupán emlékeztető jellegű szám mutatja a frekvencia — egyébként hallható — növelését. A jelsebesség alapegysége a ,,ti" jel megszólalási ideje. Ezt az értéket tudjuk növelni .05 sec-os lépésekben .5 sec-ig. A kijelzés másodpercben történik. A jelarányok:

,,ti"	1	egység
,,tá"	2	egység
jelköz	1,5	egység
betűköz	3	egység
szóköz	6	egység

Az egyes morse-jelek ,,maszkja" a datasorokban látható. Az 1-esek a ,,ti" hangot, a 2-esek a ,,tá"-t reprezentálják. A program a ,,maszkokat" sztringváltozóként kezeli, és karakterenként értékeli. A VAL függvénnyel előállított érték, a megszólalási időt határozza meg.

A program az összes alfanumerikus karakteren kívül, az alábbi írásjeleket tudja kezelni: pont, vessző, kérdőjel, kettőspont, pontosvessző, kötőjel, törtjel, zárójel.

Tekintve, hogy a használt karakterkészlet az ASCII 39 és 90 közötti tartományban van, akadnak olyan karakterek, melyeknek nincsen morse-kódja. Ha egy ilyen billentyűt nyomunk meg, a ,,tévedés" morse-jelet halljuk.

A morse-ábc nem tesz különbséget a bal és jobboldali zárójel között, így a program sem tudja ezt megtenni. Mindkét (kerek) zárójelnek ugyanaz a morse-kódja.

A programot — a gyakorláson túlmenően — ajánlani tudom a rádióamatőröknek! Ha a számítógép és adójuk modulátorfokozata közé megfelelő interface-t építenek, máris megkönnyítik munkájukat. Pl. a lemezrevett körözvényeket bármikor, bármilyen sebességgel leadhatják.

Szász Sándor

```
0 REM MORSEGYAKORLO
10 :
20 POKE 808,234
30 A=54272:B=56577:POKE 56579,1:DIM A$(51)
40 POKE 53280,8:POKE 53281,8:PRINT CHR$(158)CHR$(142)CHR$(8)
50 FOR L=1 TO 51:READ A$(L):NEXT L:T=1:GOTO 90
60 TI$="000000":W=X*(T*.3)
70 IF TI<W THEN 70
80 RETURN
90 PRINT"███  * * * * * * * * * * * * * * * * * * *"
100 PRINT"██         █ MORSE GYAKORLO PROGRAM "
110 PRINT"██     LEMEZZEL FOG DOLGOZNI ? █I█GEN/█N█EM"
120 GET W$:IF W$<>"I" AND W$<>"N" THEN 120
130 Q=1:IF W$="N" THEN 190
140 PRINT"██        IRAS LEMEZRE             <F-1>"
150 PRINT"██        OLVASAS LEMEZROL         <F-3>"
160 GET W$:IF W$<>CHR$(133) AND W$<>CHR$(134) THEN 160
170 IF W$=CHR$(133) THEN Q=2:GOTO 390
180 Q=3:POKE 214,9:SYS 58640
190 PRINT"██          HANGMAGASSAG          <F-1>"
200 PRINT"██          ADASSEBESSEG          <F-3>"
210 PRINT"██          O.K.                <SPACE>"
220 PRINT"██          E N D                 <F-8>"
230 PRINT"██  * * * * * * * * * * * * * * * * * * *"
240 GET W$:IF W$<>CHR$(133) AND W$<>CHR$(140) THEN 240
250 IF W$=CHR$(140) THEN END
260 IF P>29 THEN 280
270 POKE A+24,15:POKE A+6,240:P=30
280 POKE A+4,33:POKE 211,34:POKE 214,11:SYS 58640:PRINT (P-20)/10;" "
290 POKE A+1,P:IF P=82 THEN P=30
300 GET W$:IF W$<>CHR$(133) AND W$<>CHR$(32) THEN 300
310 IF W$=CHR$(133) THEN P=P+2:GOTO 280
320 POKE A+4,32:POKE 198,0
330 GET W$:IFW$<>CHR$(134) THEN 330
```

```
340 IF T>10 THEN T=1
350 POKE 211,34:POKE 214,14:SYS 58640:PRINT STR$(T/20);" "
360 Z=1:X=1:GOSUB 520
370 GET W$:IF W$<>CHR$(134) AND W$<>CHR$(32) THEN 350
380 IF W$=CHR$(134) THEN T=T+1:GOTO 340
390 Z=0:PRINT"██":IF Q=1 THEN 410
400 INPUT"█       A FILE NEVE: ";C$
410 IF Q<3 THEN PRINT"█ KEREM A SZOVEGET:"
420 PRINT:ON Q GOTO 430,590,610
430 POKE 204,0
440 GET W$:IF W$="" THEN 440
450 IF PEEK(207) THEN 450
460 POKE 204,1
470 PRINT W$;:IF W$=CHR$(32)THEN X=3:GOSUB 60:GOTO 560
480 IF W$=CHR$(140) THEN GOTO 570
490 D=ASC(W$)-39:IF D<1 OR D>51 THEN 430
500 IF Q=2 THEN 560
510 FOR I=1 TO LEN(A$(D)):A$=MID$(A$(D),I,1):X=VAL(A$)
520 POKE A+4,33:POKE B,1:GOSUB 60
530 POKE B,0
540 POKE A+4,32:X=1.5:GOSUB 60:IF Z THEN RETURN
550 NEXT:X=1.5:GOSUB 60
560 ON Q GOTO 430,600,620
570 IF Q>1 THEN CLOSE2
580 GOTO 90
590 OPEN2,8,2,"@:"+C$+",S,W":GOTO 430
600 PRINT#2,W$;:GOTO 430
610 OPEN2,8,2,C$+",S,R"
620 IF ST<>0 THEN 640
630 GET#2,W$:GOTO 470
640 CLOSE 2:PRINT:PRINT"██ V E G E !          <F-1>"
650 PRINT"█ ISMETLES:          <F-3>"
660 GET W$:IF W$="" THEN 660
670 IF W$=CHR$(133) THEN  90
680 IF W$=CHR$(134) THEN  PRINT"███:GOTO 610
690 GOTO 660
700 DATA 212212,212212,1111111,1111111
710 DATA 221122,211112,121212,21121
720 DATA 22222,12222,11222,11122,11112
730 DATA 11111,21111,22111,22211,22221
740 DATA 222111,212121,1111111,1111111
750 DATA 1111111,112211,1111111
760 DATA 12,2111,2121,211,1,1121,221,1111,11,1222
770 DATA 212,1211,22,21,222,1221,2212,121,111,2
780 DATA 112,1112,122,2112,2122,2211

READY.
```

BEFŰZŐ

AMIGA

Ez a kis ,,nyúlfarknyi" program egy igen hasznos kényelmi funkciót valósít meg.

Valószínűleg sokaknak eszébe jutott, hogy milyen jó is lenne, ha az ,,user" programlemezünkön a legtöbbet használt program (pl.: X-Copy) közvetlenül a lemez betétele után induljon és ne keljen végigvárni, amíg visszakapjuk a vezérlést.

Ez a program a startup file megfelelő helyére befűzve megvalósítja a fenti kívánalmakat.

A listában az Execute utasítás után a ⟨program⟩ felírat helyére kell beírnunk a sokszor használt program nevét. Ha mindent elvégeztünk és boot-oljuk a lemezt, akkor a bal egérgomb nyomvatartásával még a vezérlés visszakerülése előtt elindíthatjuk a kívánt programot.

```
/*
        BEFUZO

        AztecC: cc +l <file>
                ln <file>.o -lc32 -lm -o

              Code by TPE
                                                      */

main()
{
        int     k;

        k=(int) peekb (0xbfe001);
        if(k==188) Execute("<program>",0,0);
}
```

416×200

Régi commodore-s vagyok, bár a C64-es programozásával már csak néha foglalkozom, de egy februári újságcikkük alapján támadt egy ötletem, amit megvalósítva és sok sikert kívánva küldök commodore-osoknak.

A program kikapcsolja a jobb és bal oldali keretet (bordert), így ezt a területet nagyfelbontású grafikának lehet felhasználni, ezáltal megnövelve a vízszintes felbontást.

Ezt úgy éri el, hogy a gépikódú rutin a videó chippel (VIC) szigorúan szinkronban futva 20 raszter soronként tologat függőlegesen 2—2 sprite-ot a keret jobb és bal oldalán. Mivel hogy a képernyő függőleges felbontása 200 rasztersor, a sprite-é pedig 21, így összesen 10-szer kerül sor a 4 sprite eltolására, másodpercenként 50-szer. Ez azt jelenti, hogy összesen 40 sprite-ra van szükség ahhoz, hogy fedni tudjuk a keretet. Ezekkel a ,,4 sprite-okkal" nő meg a vízszintes felbontás 320-ról, elméleti 416-ra (320+4*24). Azért elméleti, mert a különböző megjelenítők más és más szélességben használják a képcsövet, így gyakorlatban 416 helyett kb. 380-as vízszintes felbontást lehet elérni. (Tudomásom szerint ez az első ilyen program, de ha valaki rendelkezik vagy tud hasonlót azt nagyon szívesen fogadnám.)

A program használata a gépikódú rész betöltésével kezdődik, amit vagy a DATA sorokból vagy az assembler forrásból lehet bevinni. A program SYS 49152-vel indul, de ahhoz, hogy a ,,széles" felbontású képernyőt használni tudjuk, még be kell állítani BASIC-ból (v. gépikódból) a VIC-et HIRES üzemmódba, majd a VIC báziscímét a CIA#II-vel eltolni $4000-re (16384), továbbá be kell állítani a színeket és törölni a bittérkép memóriáját. A sorrend mindegy, de azt megjegyezném, hogy a SYS 49152 kiadása után a programok futása 1/3-ára csökken a megszakítások miatt. A beállításra példa a GYALOGSZINUSZ II nevű BASIC prg. ,,GRAPIC MODE" nevű subrutinja. Visszatérni karakteres üzemmódba ugyanígy lehet lásd a ,,CARACTER MODE" nevű subrutint. Pontok rajzolásához a ,,PLOT" nevű subrutin adhat algoritmus ötletet. Persze aki gondolja megvalósíthatja gépikódban is ezt a 3 subrutint.

A program remélem megnyerte tetszésüket, használatához sok sikert kívánok.

Einhorn Róbert

```
1000           100      ;*********** WIDE-SCREEN ************
002E           110 RAS  =$2E          ; RASTER
1000           120      ;
C000           130      *=$C000       ; SYS 49152
C000 78        140      SEI           ; NEW IRQ
C001 A9 7F     150      LDA #$7F
C003 8D 0D DC  160      STA $DC0D
C006 A9 01     170 NEW  LDA #$01
C008 8D 19 D0  180      STA $D019
C00B 8D 1A D0  190      STA $D01A
C00E A9 2E     200      LDA #RAS
C010 8D 12 D0  210      STA $D012
C013 AD 11 D0  220      LDA $D011
C016 29 60     230      AND #$60
C018 09 1B     240      ORA #$1B
C01A 8D 11 D0  250      STA $D011
C01D A9 3B     260      LDA #IRQ<
C01F A0 C0     270      LDY #IRQ>
C021 8D 14 03  280 VEC  STA $0314
C024 8C 15 03  290      STY $0315
C027 58        300      CLI
C028 60        310      RTS
C029           320      ;
C029 78        330 OLD  SEI           ; OLD IRQ
C02A A9 00     340      LDA #$00
C02C 8D 1A D0  350      STA $D01A
C02F A9 81     360      LDA #$81
C031 8D 0D DC  370      STA $DC0D
C034 A9 31     380      LDA #$31
C036 A0 EA     390      LDY #$EA
C038 4C 21 C0  400      JMP VEC
C03B           410      ;
C03B A0 2F     420 IRQ  LDY #RAS+1; 2  IRQ PROGRAM
C03D 8C 12 D0  430      STY $D012     ; 4
C040 A9 51     440      LDA #IRQ2<    ; 2
C042 8D 14 03  450      STA $0314     ; 4
C045 8D 19 D0  460      STA $D019     ; 4
C048 58        470      CLI           ; 2=18TC
C049 EA        480      NOP
```

```
C04A EA          490        NOP
C04B EA          500        NOP
C04C EA          510        NOP
C04D 78          520        SEI
C04E 4C 31 EA    530        JMP $EA31        ; ROM IRQ START
C051 A2 04       540 IRQ2   LDX #$04         ;
C053 CA          550 WAIT   DEX              ; 4*5+1=21
C054 D0 FD       560        BNE WAIT         ;
C056 CC 12 D0    570        CPY $D012        ; 4
C059 F0 00       580        BEQ SYNCR        ; 2=27 OR 28TC
C05B             590        ;                ; SYNCRON TO VIC
C05B A9 E0       600 SYNCR  LDA #$E0
C05D 8D 08 D0    610        STA $D008        ; SPRITE4-X
C060 A9 00       620        LDA #$00
C062 8D 0A D0    630        STA $D00A        ; SPRITE5-X
C065 A9 58       640        LDA #$58
C067 8D 0C D0    650        STA $D00C        ; SPRITE6-X
C06A A9 70       660        LDA #$70
C06C 8D 0E D0    670        STA $D00E        ; SPRITE7-X
C06F             680        ;
C06F A9 D0       690        LDA #$D0
C071 8D 10 D0    700        STA $D010        ; SPRITE764-XHI
C074             710        ;
C074 A9 9A       720        LDA #$9A
C076 A0 04       730        LDY #$04
C078 38          740        SEC
C079 99 FB 6F    750 SPG    STA $6FFB,Y
C07C E9 0A       760        SBC #$0A         ; SPRITE-PAGE
C07E 88          770        DEY
C07F D0 F8       780        BNE SPG
C081             790        ;
C081 A9 F0       800        LDA #$F0
C083 8D 15 D0    810        STA $D015        ; SPRITE7654-ENABLE
C086             820        ;
C086 AE 16 D0    830        LDX $D016
C089 A9 1D       840        LDA #$1D         ; 2
C08B 8D 09 D0    850        STA $D009        ; 4
C08E A9 31       860        LDA #$31         ; 2
C090 D5 00       870        CMP $00,X        ; 4
C092 D5 00       880        CMP $00,X        ; 4=16TC
C094             890        ;
C094 EA          900 SPY    NOP              ; 2
C095 18          910        CLC              ; 2
C096 69 14       920        ADC #$14         ; 2
C098 8D 1B D0    930        STA $D01B        ; 4(TEMPORARY MEMEORY)
C09B AD 09 D0    940        LDA $D009        ; 4=14TC  SPRITE4-Y
C09E 8C 16 D0    950        STY $D016
C0A1 8E 16 D0    960        STX $D016        ; VERTIKAL BORDER OFF
C0A4 69 15       970        ADC #$15
C0A6 8D 09 D0    980        STA $D009        ; SPRITE4-Y
C0A9 8D 0B D0    990        STA $D00B        ; SPRITE5-Y
C0AC 8D 0D D0   1000        STA $D00D        ; SPRITE6-Y
C0AF 8D 0F D0   1010        STA $D00F        ; SPRITE7-Y
C0B2 EE FC 6F   1020        INC $6FFC        ; SPRITE4-PAGE
C0B5 EE FD 6F   1030        INC $6FFD        ; SPRITE5-PAGE
C0B8 EE FE 6F   1040        INC $6FFE        ; SPRITE6-PAGE
C0BB EE FF 6F   1050        INC $6FFF        ; SPRITE7-PAGE
C0BE 98         1060 CAR    TYA
C0BF 8D 16 D0   1070        STA $D016
C0C2 8E 16 D0   1080        STX $D016        ; VERTIKAL BORDER OFF
C0C5 99 16 D0   1090        STA $D016,Y
C0C8 8E 16 D0   1100        STX $D016        ; VERTIKAL BORDER OFF
C0CB AD 12 D0   1110        LDA $D012        ; RASTER REGISTER
C0CE C9 F9      1120        CMP #$F9
C0D0 D0 0E      1130        BNE JP1
C0D2 AD 11 D0   1140        LDA $D011
C0D5 29 F7      1150        AND #$F7
C0D7 8D 11 D0   1160        STA $D011        ; HOR. BORDER OFF
C0DA C1 00      1170        CMP ($00,X)      ; 6
C0DC C1 00      1180        CMP ($00,X)      ; 6=12TC
```

```
C0DE D0 DE       1190         BNE CHR
C0E0 C9 FB       1200 JP1     CMP #$FB   ; SCREEN END ANALYZATION
C0E2 D0 02       1210         BNE JP2
C0E4 F0 13       1220         BEQ END
C0E6 CD 1B D0    1230 JP2     CMP $D01B  ; SPRITE CHANGE ANALYZATION
C0E9 F0 A9       1240         BEQ SPY
C0EB 38          1250         SEC
C0EC E9 01       1260         SBC #$01
C0EE 4D 11 D0    1270         EOR $D011
C0F1 29 07       1280         AND #$07
C0F3 D0 C9       1290         BNE CHR            ; CARACTER-LINE-DATA READING ANALYZAT
ON
C0F5 48          1300         PHA                ; 3
C0F6 68          1310         PLA                ; 4=7TC
C0F7 F0 C5       1320         BEQ CHR
C0F9 20 06 C0    1330 END     JSR NEW
C0FC 4C 81 EA    1340         JMP $EA81          ; ROM IRQ END
C0FF             1350         ;
C0FF             1360         .END
```

```
LINES:127   SYMBOLE:14   ERRORS:0

CAR   =C0BE  END   =C0F9  IRQ   =C03B  IRQ2  =C051  JP1   =C0E0  JP2   =C0E6
NEW   =C006  OLD   =C029  RAS   =002E  SPG   =C079  SPY   =C094  SYNCR =C05B
VEC   =C021  WAIT  =C053
```

```
1 REM **********************************
2 REM * ----- DATA WIDE-SCREEN ------ *
3 REM * PROGRAMOT IRTA:EINHORN ROBERT *
4 REM *              /HUNGARIA/       *
5 REM *              1991.04./        *
6 REM **********************************
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES...";:J=49152:VE=49407:P=J
9 FOR B=0 TO 7:READ A$
10 L=ASC(MID$(A$,2,1))
11 H=ASC(MID$(A$,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"▮▮▮▮▮▮▮▮▮"P;:P=P+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ... SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"▮▮▮KESZ":END
20 DATA 78,A9,7F,8D,0D,DC,A9,01,0960
21 DATA 8D,19,D0,8D,1A,D0,A9,2E,0964
22 DATA 8D,12,D0,AD,11,D0,29,60,0902
23 DATA 09,1B,8D,11,D0,A9,3B,A0,0790
24 DATA C0,8D,14,03,8C,15,03,58,0608
25 DATA 60,78,A9,00,8D,1A,D0,A9,0929
26 DATA 81,8D,0D,DC,A9,31,A0,EA,1115
27 DATA 4C,21,C0,A0,2F,8C,12,D0,0874
28 DATA A9,51,8D,14,03,8D,19,D0,0788
29 DATA 58,EA,EA,EA,EA,78,4C,31,1269
30 DATA EA,A2,04,CA,D0,FD,CC,12,1285
31 DATA D0,F0,00,A9,E0,8D,08,D0,1198
32 DATA A9,00,8D,0A,D0,A9,58,8D,0926
33 DATA 0C,D0,A9,70,8D,0E,D0,A9,1033
34 DATA D0,8D,10,D0,A9,9A,A0,04,1060
35 DATA 38,99,FB,6F,E9,0A,88,D0,1158
36 DATA F8,A9,F0,8D,15,D0,AE,16,1223
37 DATA D0,A9,1D,8D,09,D0,A9,31,0982
38 DATA D5,00,D5,00,EA,18,69,14,0809
39 DATA 8D,1B,D0,AD,09,D0,8C,16,0928
40 DATA D0,8E,16,D0,69,15,8D,09,0856
41 DATA D0,8D,0B,D0,8D,0D,D0,8D,1071
42 DATA 0F,D0,EE,FC,6F,EE,FD,6F,1426
43 DATA EE,FE,6F,EE,FF,6F,98,8D,1500
44 DATA 16,D0,8E,16,D0,99,16,D0,0985
45 DATA 8E,16,D0,AD,12,D0,C9,F9,1221
46 DATA D0,0E,AD,11,D0,29,F7,8D,1049
47 DATA 11,D0,C1,00,C1,00,D0,DE,1041
48 DATA C9,FB,D0,02,F0,13,CD,1B,1153
49 DATA D0,F0,A9,38,E9,01,4D,11,1001
50 DATA D0,29,07,D0,C9,48,68,F0,1081
51 DATA C5,20,06,C0,4C,81,EA,00,0866

READY.
```

```
100 REM ******* GYALUUSZINUSZ II *******
110 HT=0:SZ=11:GOSUB 250:REM GRAFIK MOD
120 :
130 Y=99:FOR X=0 TO 415:GOSUB 480:NEXT
140 X=208:FOR Y=0 TO 199:GOSUB 480:NEXT
150 X=0:FOR I=0 TO 2*π STEP π/208
160 Y=100+99*SIN(I):GOSUB480:X=X+1:NEXT
170 :
180 POKE 198,0:WAIT 198,1
190 GOSUB 380:REM KARAKTER MOD
200 END
210 :
220 :
230 :
240 :
250 REM -----(GRAPHIC MODE)-----------
260 SYS 49152
270 POKE 56578,PEEK(56578) OR 3
280 POKE 56576,(PEEK(56576) AND 252) OR 2:REM      ( VICBAZISCIM=16384 )
290 POKE 53265,PEEK(53265) OR 32:REM               ( GRAFIK UZEMMOD )
300 VM=PEEK(53272):POKE 53272,11*16:REM            ( BITMAP: 16384-24383 )
310 FOR I=16384+11*1024 TO 16384+12*1024-25:POKE I,HT+SZ*16:NEXT:REM KEP SZINEI
320 RE=PEEK(53280):REM KERETSZIN TAROLAS
330 POKE 53280,HT:FOR I=0 TO 3:POKE 53291+I,SZ:NEXT:REM   ( A KERET SZINEI )
340 FOR I=16384 TO 26943:POKE I,0:NEXT:REM            ( BIT TERKEP TORLESE )
350 RETURN
360 :
370 :
380 REM ---------(CHARACTER MODE)-------
390 POKE 53280,RE
400 POKE 53265,PEEK(53265) AND (255-32)
410 POKE 56578,PEEK(56578) OR 3
420 POKE 56576,(PEEK(56576) AND 252) OR 3
430 POKE 53272,VM
440 SYS 49193
450 RETURN
460 :
470 :
480 REM ---------(PLOT RUTIN)-----------
490 X=INT(X+.5):Y=INT(Y+.5)
500 IF X<0 OR X>415 OR Y<0 OR Y>199 THEN RETURN
510 IF X>47 AND X<368 THEN C=16384+40*(YAND248)+((X-48)AND504)+(YAND7):GOTO 540
520 R=INT(X/8):R=R-40*SGN(R AND 32)
530 L=INT(R/3):C=24384+640*L+R-3*L+64*INT(Y/20)+3*Y-63*INT(Y/21)
540 POKE C,PEEK(C) OR 2↑(7-(X AND 7)):RETURN

READY.
```

OTP-HITEL

1. A program kezelése:
A menüpontok alapján, a kérdésekre kell válaszolni.

2. A program célja:
OTP kamatozású kölcsönök számítása, vizsgálata.

3. A program műfaja: felhasználói program.

4. A program leírása:
— A program segítségével, tetszőlegesen megadott adatokkal a havi törlesztés számítható. A kért adatok: a hitel éve, a hitel összege, hitel kamata, törlesztő évek száma. Ezek az adatok lehetnek a régen megkötött szerződés jelenlegi adatai is. Így a hitel éve: 1992-nek, a hitel összege: az évzáráskor fennmaradt összeggel egyezik meg. A hitel kamata: a jelenlegi 32%, a törlesztő évek száma: a fent maradt éveknek felel meg. A program két fajta képpen számolja ki a törlesztés összegét, így a programozó képernyőn ellenőrizheti a számok helyességét. A képernyő három részre van osztva.

1. Felső részen a kért adatok jelennek meg.

2. A középső részen a gép által számolt adatok:
— a hitel évenkénti kamatai (kamatos kamat)
— a hitel lejártakor fennmaradt hitelmaradék ennek általában nullának kell lennie)
— a kamat összege, az évek során számolt kamatok összege
— a visszafizetés összege, a kamatok és a hitel összege. (Ennek meg kell egyeznie a törlesztés összegével.).

3. Alsó részen a képlettel számolt adatok:
A képlet a középiskolákban használatos függvénytáblázatból lett kölcsönözve (t kölcsön törlesztéshez szükséges évi részlet)
évi törlesztés
— havi törlesztés: évi törlesztés lebontása
— törlesztés összege: évi törlesztés és a törlesztő évek számával számolva.
— A program tetszőleges adatokkal táblázatot készít. A táblázat nagyon hasznos lehet, további törlesztésünk figyelésénél.

A kért adatok: hitel összege és a hitel kamata.

A gép ezután elkészíti a táblázatot, melyből a következőket lehet kiolvasni.

Egy választott havitörlesztés esetén, a felvett hitelt mennyi év alat törleszthetjük. Továbbá a felvett hitel, a kamatokkal együtt mennyire „duzzadt meg". (Pontosabban a hiteltörlesztés összegét.) Látható a táblázatból, hogy a törlesztő évek növekedésével a havi törlesztés csökken. Hosszú távon a havi törlesztés nagysága meg is áll. De a hiteltörlesztés összege többszöröse a felvett hitelnek (kamatok miatt).

A megoldás a hitelektől a lehetőségekhez mérten, minnél hamarabb megszabadulni. Havi 100 forintok esetén is éveket lehet nyerni. A kapott táblázatot kívánság szerint, ki is lehet nyomtatni (annyi példányban, amennyiben kívánja a felhasználó).

5. Programozási feladatok megoldása:
— matematikai képlet alkalmazása
— táblázat nyomtatása

6. Megjegyzés:
— A programmal kapott adatokat ne vegye senki 100%-nak, nem biztos, hogy megegyezik az OTP adataival. Az OTP különálló rendszer, nem biztos, hogy érdemes vele fillérekért veszekedni. A kapott adatok arra viszont nagyon jók, hogy tájékozódjunk.

```
AZ 5-RE VEGZODO SORSZAMU SOROKAT NEM KELL BEIRNI!

10 REM---COMMODORE GEPEKRE-----------
20 REM ***************************
30 REM * OTP-HITEL               *
40 REM * KESZITETTE:             *
50 REM *          LOOS JOZSEF    *
60 REM * COMMODORE PLUS/4 GEPEN. *
70 REM ***************************
80 REM -------------------------
90 PRINT"":DIM C(36)
95 REM [CLR]
100 PRINT"O T P KAMATOZASU"
105 REM 5X[LE] 12X[JOBB]
110 PRINT"KOLCSONOK"
115 REM [LE] 15X[JOBB]
120 PRINT"------------------------------------"
125 REM 2X[LE] 2X[JOBB] 36X[-]
130 PRINT"-SOOLSOFT-"
135 REM 5X[LE] 14X[JOBB]
140 PRINT"(1991)"
145 REM [LE] 16X[JOBB]
150 FOR WW=1 TO 1000:NEXT WW
160 PRINT""
165 REM [CLR]
```

```
170 PRINT"1. ON ALTAL MEGADOTT ADATOKKAL, KOL-":PRINT"CSON TORLESZTES ";
175 REM [LE] [JOBB] 4X[JOBB]
180 PRINT"SZAMITASA."
190 PRINT"2. TETSZ. ADATOKKAL TABLAZAT KESZITESE"
195 REM [LE] [JOBB]
200 PRINT"3. VEGE."
205 REM [LE] [JOBB]
210 PRINT"----------------------------------------"
215 REM [LE] 40X[-]
220 PRINT"(VALASSZON 1-3 !)
225 REM 6X[LE] 3X[JOBB]
230 PRINT" TOVABB,LAPOZAS-BARMELY BILL.-VEL "
235 REM [LE] 3X[JOBB] [CTR/9] [CTR/0]
240 GET A$:IF A$="" THEN GOTO 240
250 IF A$="1" THEN J=4:GOTO 310
260 IF A$="2" THEN GOTO 650
270 IF A$="3" THEN PRINT"":END
275 REM [CLR]
280 IF A$<>"1" OR A$<>"2" OR A$<>"3" THEN GOTO 240
290 REM ********** 1. ***************
300 REM --------------------------------
310 INPUT "HITEL EVE:";EV
315 REM [CLR]
320 INPUT "HITEL OSSZEGE:";OS
330 INPUT "HITEL KAMATA:";KA
340 INPUT "TORLESZTO EVEK SZAMA :";EK:VE=EV+EK
350 PRINT"----------------------------------------"
355 REM 40X[-]
360 REM --------------------------------
370 PRINT""
375 REM [CLR]
380 PRINT"HITEL FELVETEL EVE:";EV
390 PRINT"HITEL LEJARASANAK EVE:";VE
400 PRINT"HITEL KAMATA:";KA;"%"
405 REM [CTR/9] [CTR/0]
410 PRINT"HITEL OSSZEGE:";OS
420 PRINT"========================================"
425 REM 40X[=]
430 GOSUB 610:GOSUB 540:PRINT""
435 REM [LE]
440 PRINT"MARADEK HITEL:";INT(E)
450 PRINT"KAMAT OSSZEGE:";INT(KO)
460 PRINT"VISSZAFIZETES OSSZEGE:";OS+INT(KO)
470 PRINT"----------------------------------------"
475 REM 40X[-]
480 PRINT"HAVI TORLESZTES:";A/12
490 PRINT"EVI TORLESZTES:";A
500 PRINT"TORLESZTES OSSZEGE:";A*EK
510 GET A$:IF A$="" THEN GOTO 510
520 GOTO 160
530 REM --------------------------------
540 KO=0:E=OS
550 FOR Q=1 TO EK
560 C(Q)=E*(KA/100)
570 PRINT INT(C(Q)),
580 KO=KO+C(Q)
590 E=E+C(Q)-A
600 NEXT Q:RETURN
610 REM --------------------------------
620 A=OS*((1+KA/100)↑EK)*((KA/100)/(((1+KA/100)↑EK)-1))
630 RETURN
640 REM *********** 2. ***************
650 REM --------------------------------
```

```
660 PRINT"□":INPUT"▒HITEL OSSZEGE:";OS
665 REM [CLR] [LE] [JOBB]
670 INPUT"▒HITEL KAMATA:";KA:PRINT"□"
675 REM [LE] [JOBB] [CLR]
680 PRINT OS;,KA;"%"
690 PRINT"----------------------------------------"
695 REM 40X[-]
700 PRINT" EV-HAVI-OSSZBEFIZ.";
710 PRINT" HAVI-OSSZBEFIZ.-EV"
720 PRINT"----------------------------------------"
725 REM 40X[-]
730 FOR EK=1 TO 18
740 GOSUB 860:AA=A
750 GOSUB 890:BB=A
760 PRINT EK;TAB(4)INT(AA/12);TAB(11)INT(AA*EK);TAB(20)INT(BB/12);
770 PRINT TAB(26)INT(BB*(EK+18));TAB(32)EK+18
780 NEXT EK
790 PRINT TAB(6)"▒KER NYOMTATAST (I/N)?▒"
795 REM [CTR/9] [CTR/0]
800 GET A$:IF A$="" THEN GOTO 800
810 IF A$="I" THEN GOTO 930
820 IF A$="N" THEN GOTO 840
830 IF A$<>"I" OR A$<>"N" THEN GOTO 800
840 GOTO 160
850 REM -----------------------------------
860 A=OS*((1+KA/100)↑EK)*((KA/100)/(((1+KA/100)↑EK)-1))
870 RETURN
880 REM -----------------------------------
890 A=OS*((1+KA/100)↑(EK+18))*((KA/100)/(((1+KA/100)↑(EK+18))-1))
900 RETURN
910 REM ---------------------------
920 REM ******* NYOMTATAS **********
930 PRINT"□":PRINT "      ▒KER NYOMTATAST (I/N)?▒"
935 REM 2X[FEL] 6X[SPC] [CTR/9] [CTR/0]
940 OPEN 1,4:TB$=CHR$(16)
950 PRINT#1,"----------------------------------------"
955 REM 40X[-]
960 PRINT#1,OS;"       ";KA;"%"
965 REM 7X[SPC]
970 PRINT#1,"----------------------------------------"
975 REM 40X[-]
980 PRINT#1," EV-HAVI-OSSZBEFIZ.";
990 PRINT#1," HAVI-OSSZBEFIZ.-EV"
1000 PRINT#1,"----------------------------------------"
1005 REM 40X[-]
1010 FOR EK=1 TO 18
1020 GOSUB 860:AA=A
1030 GOSUB 890:BB=A
1040 PRINT#1,EK;"  "INT(AA/12);TB$"11"INT(AA*EK);TB$"20"INT(BB/12);
1050 PRINT#1,TB$"26"INT(BB*(EK+18));TB$"32"EK+18
1060 NEXT EK
1070 PRINT#1
1080 CLOSE 1:PRINT"□KESZ !";:GOTO 790
1085 REM [FEL]
```

```
HITEL FELVETEL EVE: 1992
HITEL LEJARASANAK EVE: 1996
HITEL KAMATA:▒▒ %
HITEL OSSZEGE: 120000
====================================
  38400     32364     24397     13881
```

```
MARADEK HITEL: 0
KAMAT OSSZEGE: 109043
VISSZAFIZETES OSSZEGE: 229043
------------------------------------------
HAVI TORLESZTES: 4771.74184
EVI TORLESZTES: 57260.902
TORLESZTES OSSZEGE: 229043.608


 12000     32 %
------------------------------------------
 EV-HAVI-OSSZBEFIZ. HAVI-OSSZBEFIZ.-EV
------------------------------------------
1   1319   15839    321   73335   19
2   751    18024    321   77098   20
3   566    20381    320   80877   21
4   477    22904    320   84668   22
5   426    25584    320   88469   23
6   394    28410    320   92277   24
7   373    31373    320   96092   25
8   358    34458    320   99913   26
9   348    37654    320   103737   27
10  341    40949    320   107565   28
11  335    44331    320   111395   29
12  331    47787    320   115227   30
13  328    51309    320   119061   31
14  326    54885    320   122897   32
15  325    58509    320   126733   33
16  323    62171    320   130570   34
17  322    65867    320   134408   35
18  322    69590    320   138246   36
```

JACKIE UTILITES C-64

I. rész

Te, kedves Commodore-64 rajongó, a továbbiakban a ,,JACKIE UTILITIES" azon programjait láthatod, amelynek a programcsomagja az előző közlés óta elkészült. Bizonyára találni fogsz köztük olyat, amelyet rutinkönyvtáradba felvéve, valamely programodban majdan alkalmazni tudsz. Mindennemű felhasználás engedélyezett! És most lássuk a termést!

Az 1. listán a ,,SPRITE POSITIER" nevű rutint láthatod, ami a szellemek mozgatásához nyújt segítséget. Használata: *SYS 49152, sprite szám, x % (0), y % (0), lassítás.* A sprite-szám egyértelműen: 0—7 lehet. A lassítást 0 és 255 között lehet változtatni, de nulla értéknél kikapcsolódik a rutin. A két (egyébként tetszőleges nevű) tömbnek egész típusú, egydimenziós és azonos elemszámra (max. 255!) dimenzionáltnak kell lennie. Az elsőként megadottba az x, a másodikként jelöltbe pedig az y koordináta-értékeket tárojuk. A rutin az IRQ-hoz hozzátűződik és rendre a lassításnak megfelelő számú megszakításban a megadott szellemet a rendre következő tömbelemekben tárolt helyre pozícionálja. Ha a tömbök feltöltéséhez pl. SIN és COS függvényeket alkalmazunk, egyszerűen lehetővé válik a sprite mozgatása, ellipszis pályán vagy pl. egy színusz-hullám mentén.

A 2. lista programja szintén az objektumok kezelését könnyíti meg. Segítségével animálhatóvá válnak sprite-jaink. A ,,SPRITE AMINATE"-ot a következő parancssal indíthatjuk: *SPRITE 49152, sprite-szám, késleltetés, sprite blokk, s-blokk 2, …, s-blokk n.* A program funkciója egyértelmű: a megadott szellemhez a késleltetésnek megfelelő időközönként a soron következő sprite-blokkot rendeli hozzá. A leállításhoz a késleltetés mértékét nullában határozzuk meg.

A memóriában lévő szellemeket nyolcszoros nagyításban a képernyő tetszőleges helyére rajzoltathatjuk a ,,SPRITE NAGYÍTÓ"-val (3). Indítása: *SYS 49152, x, y, szín, sprite-blokk, kikapcsolt pontok ASCII, bekapcsolt pontok ASCII.* Az x; y a képernyő egy pontja, ahová a sprite bal felső sarka kerül. Meg kell adnunk a rajzolás színét (0—15), a sprite-blokkot, valamint annak a két karakternek az ASCII kódját, amit a be-, illetve a kikapcsolt pontoknál látni óhajtunk.

```
100 REM"┌──────────────────────┐"
110 REM"|    SPRITE POSITIER   |"
120 REM"|     COMMODORE  64    |"
130 REM"├──────────────────────┤"
140 REM"|IRTA: JACKIE MAESTRO  |"
150 REM"└──────────────────────┘"
160 :
170 K=49152:V=49535
180 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*
(MID$(A$,X,1)>"@")
190 FOR I=K TO V STEP 8:S=0
200 ::FOR J=0 TO 7
210 :::READ A$
```

```
220 :::A=16*FNA(1)+FNA(2)
230 :::POKE I+J,A:PRINT"]";I+J
240 :::S=S+A
250 ::NEXT
260 ::READ A$
270 ::A=16*FNA(1)+FNA(2)
280 IF(SAND255)=ATHENNEXT:PRINT"OK":END
290 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
300 DATA 20,FD,AE,20,9E,B7,E0,08,28
301 DATA 90,03,4C,48,B2,8A,0A,85,F2
302 DATA 8B,20,FD,AE,20,8B,B0,38,E9
303 DATA A5,47,E9,07,85,F7,A5,48,45
304 DATA E9,00,85,F8,20,FD,AE,20,51
305 DATA 8B,B0,38,A5,47,E9,07,85,D4
306 DATA F9,A5,48,E9,00,85,FA,A0,EE
307 DATA 00,B1,F7,0A,B0,05,A2,16,1F
308 DATA 4C,3A,A4,B1,F9,0A,90,F6,64
309 DATA C8,B1,F7,0A,90,F0,B1,F9,A4
310 DATA 0A,90,EB,A0,04,B1,F7,C9,9A
311 DATA 01,F0,05,A2,12,4C,3A,A4,D4
312 DATA B1,F9,C9,01,D0,F5,A0,05,DE
313 DATA B1,F7,C9,00,D0,ED,C8,B1,A7
314 DATA F7,85,8C,88,B1,F9,C9,00,03
315 DATA D0,E1,C8,B1,F9,C5,8C,D0,44
316 DATA DA,18,A5,F7,69,07,85,F7,7A
317 DATA A5,F8,69,00,85,F8,18,A5,40
318 DATA F9,69,07,85,F9,A5,FA,69,EF
319 DATA 00,85,FA,A2,00,A0,01,B1,73
320 DATA F9,9D,00,C8,88,B1,F9,C9,59
321 DATA 00,F0,03,4C,48,B2,18,A5,F6
322 DATA F9,69,02,85,F9,A5,FA,69,EA
323 DATA 00,85,FA,E8,E4,8C,D0,DD,84
324 DATA A2,00,A0,01,B1,F7,9D,00,88
325 DATA C6,88,A9,00,9D,00,C7,B1,0C
326 DATA F7,C9,02,90,03,4C,48,B2,9B
327 DATA C9,00,F0,11,BD,00,C6,C9,16
328 DATA 59,B0,F2,A5,8B,4A,A8,B9,D6
329 DATA 77,C1,9D,00,C7,E8,E4,8C,F4
330 DATA F0,10,18,A5,F7,69,02,85,A4
331 DATA F7,A5,F8,69,00,85,F8,4C,C6
332 DATA C2,C0,A9,00,85,8D,20,FD,5A
333 DATA AE,20,9E,B7,86,8E,E0,00,17
334 DATA D0,0D,78,A9,31,8D,14,03,D3
335 DATA A9,EA,8D,15,03,58,60,A9,99
336 DATA 00,85,8F,78,A9,30,8D,14,06
337 DATA 03,A9,C1,8D,15,03,58,60,CA
338 DATA E6,8F,A5,8F,C5,8E,F0,03,EF
339 DATA 4C,31,EA,A9,00,85,8F,A6,CA
340 DATA 8D,A4,8B,BD,00,C6,99,00,D8
341 DATA D0,BD,00,C8,99,01,D0,98,57
342 DATA 4A,AA,BD,77,C1,49,FF,2D,5E
343 DATA 10,D0,8D,10,D0,A6,8D,BD,3D
344 DATA 00,C7,0D,10,D0,8D,10,D0,21
345 DATA E6,8D,A5,8D,C5,8C,D0,04,CA
346 DATA A9,00,85,8D,4C,31,EA,01,23
347 DATA 02,04,08,10,20,40,80,FF,FD
348 REM ==============================

READY.
```

```
100 REM"┌──────────────────────┐"
110 REM"|   SPRITE  NAGYITO    |"
120 REM"|    COMMODORE  64     |"
130 REM"├──────────────────────┤"
140 REM"|IRTA: JACKIE MAESTRO  |"
150 REM"└──────────────────────┘"
160 :
170 K=49152:V=49346
180 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*
(MID$(A$,X,1)>"@")
190 FOR I=K TO V STEP 8:S=0
200 ::FOR J=0 TO 7
210 :::READ A$
220 :::A=16*FNA(1)+FNA(2)
230 :::POKE I+J,A:PRINT"]";I+J
240 :::S=S+A
```

```
250 ::NEXT
260 ::READ A$
270 ::A=16*FNA(1)+FNA(2)
280 IF(SAND255)=ATHENNEXT:PRINT"OK":END
290 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
300 DATA 38,20,F0,FF,86,8E,84,8D,6C
301 DATA 20,FD,AE,20,9E,B7,E0,11,31
302 DATA 90,03,4C,48,B2,86,02,20,81
303 DATA FD,AE,20,9E,B7,E0,05,B0,B5
304 DATA F1,A4,02,18,20,F0,FF,18,D6
305 DATA A5,D1,65,D3,85,FB,85,FD,B0
306 DATA A5,D2,69,00,85,FC,18,69,E2
307 DATA D4,85,FE,20,FD,AE,20,9E,E0
308 DATA B7,86,F9,20,FD,AE,20,9E,BF
309 DATA B7,A9,00,85,F8,8A,A2,00,09
310 DATA 0A,26,F8,E8,E0,06,D0,F8,BE
311 DATA 85,F7,20,FD,AE,20,9E,B7,BC
312 DATA 86,FA,20,FD,AE,20,9E,B7,C0
313 DATA 86,FF,A0,00,84,02,A9,00,54
314 DATA 85,8C,A4,02,B1,F7,85,8B,6F
315 DATA E6,02,A2,00,06,8B,A5,FA,BA
316 DATA 90,02,A5,FF,A4,8C,91,FB,F2
317 DATA A5,F9,91,FD,E6,8C,E8,E0,66
318 DATA 08,D0,E9,A5,8C,C9,18,D0,A3
319 DATA D9,18,A5,FB,69,28,85,FB,A2
320 DATA A5,FC,69,00,85,FC,18,A5,48
321 DATA FD,69,28,85,FD,A5,FE,69,1C
322 DATA 00,85,FE,A5,02,C9,3F,D0,02
323 DATA B5,18,A6,8E,A4,8D,20,F0,42
324 DATA FF,60,00,00,00,00,00,00,5F
325 REM ==============================

READY.
```

```
100 REM"┌──────────────────────┐"
110 REM"|   SPRITE  ANIMATE    |"
120 REM"|    COMMODORE  64     |"
130 REM"├──────────────────────┤"
140 REM"|IRTA: JACKIE MAESTRO  |"
150 REM"└──────────────────────┘"
160 :
170 K=49152:V=49281
180 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*
(MID$(A$,X,1)>"@")
190 FOR I=K TO V STEP 8:S=0
200 ::FOR J=0 TO 7
210 :::READ A$
220 :::A=16*FNA(1)+FNA(2)
230 :::POKE I+J,A:PRINT"]";I+J
240 :::S=S+A
250 ::NEXT
260 ::READ A$
270 ::A=16*FNA(1)+FNA(2)
280 IF(SAND255)=ATHENNEXT:PRINT"OK":END
290 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
300 DATA 20,FD,AE,20,9E,B7,E0,08,28
301 DATA 90,03,4C,48,B2,86,8B,20,0A
302 DATA FD,AE,20,9E,B7,86,8C,A0,D2
303 DATA 00,84,02,A0,00,B1,7A,C9,1A
304 DATA 2C,D0,10,20,FD,AE,20,9E,95
305 DATA B7,8A,A4,02,99,00,C2,C8,0A
306 DATA 4C,19,C0,A9,00,85,F7,85,CF
307 DATA 8D,A6,8C,E0,00,D0,0D,78,F4
308 DATA A9,31,8D,14,03,A9,EA,8D,9E
309 DATA 15,03,58,60,78,A9,59,8D,07
310 DATA 14,03,A9,C0,8D,15,03,58,7D
311 DATA 60,E6,8D,A5,8D,C5,8C,F0,46
312 DATA 03,4C,31,EA,A9,00,85,8D,25
313 DATA A6,F7,BD,00,C2,A6,8B,9D,EA
314 DATA F8,07,E6,F7,A5,F7,C5,02,3F
315 DATA D0,04,A9,00,85,F7,4C,31,76
316 DATA EA,00,00,00,00,00,00,00,EA
317 REM ==============================

READY.
```

PÖTYÖGŐ SZOLGÁLAT

263	C64	91/ 2	40 Sprite	100.-
264	C64	91/ 2	Raster - Master	60.-
265	C64	91/ 3	Rulett	80.-
266	+4	91/ 3	Tape-disc copy	90.-
267	C64	91/ 3	BASIC rearranger	40.-
268	C64	91/ 3	BASIC decompactor	40.-
269	C64	91/ 3	BASIC merger	40.-
270	C64	91/ 3	SEIKOSHA	60.-
271	C64	91/ 3	Cartridge szimuláció	100.-
272	+4	91/ 3	Sally kulcsszókereső	40.-
273	+4	91/ 3	Merge C+4	60.-
274	C128	91/ 4	1.5 MHz-es C64	80.-
275	C64	91/ 4	Morze - H	60.-
276	C64	91/ 4	Gprint	40.-
277	C64	91/ 4	ASCII - CHR$	40.-
278	C64	91/ 5	Rendező	60.-
279	C64	91/ 5	Keretbeiró	50.-
280	C64 +4	91/ 5	NLQ Print	50.-
281	C64	91/ 5	FLD	40.-
282	C64	91/ 5	Raster Split	40.-
283	+4	91/ 6	Programnyilvántartó	60.-
284	C64	91/ 6	Regiszter	50.-
285	C64	91/ 6	Digitalizált zene	40.-
286	C64	91/ 6	Ékezetes MPS 802	60.-
287	C64	91/ 7	*Ékezetes iratkészitő	80.-
288	+4	91/ 7	Bűvös téglalap	80.-
289	C64	91/ 7	Dominó - H	80.-
290	C64	91/ 7	Raktáros	60.-
291	+4	91/ 7	Tologató	50.-
292	C64	91/ 7	Aren 2000	40.-
293	C64	91/ 7	Bad Lamps	60.-
294	C64	91/ 7	Operation Patriot	40.-
295	+4	91/ 7	Mozaik - B	70.-
296	+4	91/ 7	Logi Kód	50.-
297	C64	92/ 1	Databevivő	40.-
298	+4	91/ 7	Bomber	40.-
299	+4	91/ 7	Dominó - K	40.-
300	+4	91/ 7	The Wall	30.-
301	C64	91/ 7	Mozaik - K	40.-
302	+4	91/ 7	Dominó - R	50.-
303	+4	91/ 7	*Borgok Kincse	150.-
304	+4	91/ 7	Asteroids	50.-
305	C64	91/ 9	Ékezetes irat forrás	30.-
306	C64	91/ 9	Yugo2	40.-
307	C64	91/ 9	Adventure - Sz	100.-
308	C64	91/ 9	Morze Oktató B.	100.-
309	+4	91/ 9	Memory Game Plus	80.-
310	C64	91/ 9	*Capitaly	100.-
311	C64	91/ 9	*Gengszterek	50.-
312	C64	91/ 9	*Küldetés 2000	80.-
313	C64	91/ 9	Itt a $ hol a $	40.-
314	C64	91/ 9	Hewing	40.-
315	+4	91/10	Mikrobi	40.-
316	+4	91/10	Nest of Fleas	150.-
317	+4	91/10	Adventure - F	80.-
318	+4	91/10	A világ szeme	120.-
319	C64	91/10	Gondolkoz6	80.-
320	C64	91/10	Memori - B	100.-
321	+4	91/10	Matekastély	80.-
322	C64	91/11	Lottó	60.-
323	C64	91/11	Totó	60.-
324	C64	91/11	Német gyakorló	50.-
325	+4	91/11	Beszélő szótár	100.-
326	+4	91/11	File examiner	50.-
327	C64	91/11	Sprite szerkesztő	40.-
328	C64	91/11	Egyszinü Sprite	40.-
329	C64	91/12	Rulett javitás	40.-
330	C64	91/12	Tőzsde	100.-
331	C64	91/12	IQ teszt	120.-
332	C64	92/ 1	Kalandlap	60.-
333	C64	92/ 1	Levéliró	60.-
334	C64	92/ 1	Képernyőmaszkoló	100.-
335	C64	92/ 1	Sprite Kezelő	80.-
336	C64	92/ 1	SI - OPART	30.-
337	C64	92/ 2	Patience Quartet	100.-
338	C64	92/ 2	Monopoly	100.-
339	C64	92/ 2	Akasztófa	60.-
340	C64	92/ 2	Karaktertervező	50.-
341	C64	92/ 3	String Array Manager	80.-
342	C64	92/ 3	Sprite Basic	40.-
343	C64	92/ 3	Screen Help	40.-
344	C64	92/ 3	Gyors validate	50.-
345	C64	92/ 3	BASIC SYNTAXER I.	150.-
346	C64	92/ 4	Tologató C64-re	80.-
347	C64	92/ 4	ABC rendező	80.-
348	C64	92/ 5	*Matek + Római számok	80.-
349	+4	92/ 6	Tesztkészitő + értékelő	80.-
350	C64	92/ 3	BASIC SYNTAXER II.	150.-
351	+4	92/ 7	Három játék	90.-
352	+4	92/ 7	Szökés	60.-
353	C64	92/ 7	Toto	120.-
354	C64	92/ 7	DIGIT analizis program	90.-
355	+4	92/ 7	SCREEN DUMP	120.-
356	C64	92/ 9	UNISEQ filekezelő	100.-
357	C64	92/ 9	Graphic Copy	50.-
358	C64	92/10	Printfox Resizer	100.-
359	C+4	92/11	Tologató II.	120.-
360	C+4	92/12	Sprite Plus-4-en	100.-
361	C64	92/12	Változólista	120.-

A *-gal jelölt programok csak lemezen rendelhetők !

Kérem, hogy a **Commodore Újság** legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:

(minden kockába egy betűt írjon)

Tagsági szám: ....

Feladandó az újság címére:
Commodore Újság
Budapest, 1388. Pf. 86.

Alulírott megrendelem a következő programokat a PÖTYÖGŐSZOLGÁLAT-tól:

PROGRAM SORSZÁMA	ÁRA
1.	,- Ft
2.	,- Ft
3.	,- Ft
4.	,- Ft
5.	,- Ft
6.	,- Ft
7.	,- Ft
8.	,- Ft
9.	,- Ft
Összesen: _ db	,- Ft

A programokat a SZOLGÁLAT által biztosított
lemezre (99 Ft/db)
kazettára (40 Ft/db)
az általam küldött adathordozóra kérem.
(Kérjük a megfelelő szöveg aláhúzását!)
Postaköltség (35 Ft)
A fizetés módja: személyesen — csekken — utánvéttel
(Kérjük a megfelelő szöveg aláhúzását!)
ÖSSZESEN:

A megrendelő neve: ....

Címe: ....

NOVOTRADE SZERVIZ Kft.

BÚCSÚZUNK ...

... a Magyar utcai C= szerviztől

Talán éppen most van egy évtizede, hogy az első fecskének számító CBM, azaz Commodore Business Machines számítógépek elérhető áron és mondjuk úgy, hogy közforgalomban beszerezhetően megjelentek a hazai piacon, és ezzel kezdetét vette egy, a napjainkig is tartó „számítógépláz". 1992 karácsonyán is bizonyára még sokak kedves ajándéka volt „a számítógép", ami az elmúlt egy évtizedben a Commodore típusokon kívül sokfélét jelenthetett, kezdve a Sinclairtől a mai PC-gépekig.

A legnagyobb sikert kétségkívül a C 64-es aratta, hogy miért, erre mindenkinek megvan a maga válasza, de tény, hogy mind a mai napig, hála egyre olcsóbban, de veszik ... és veszik...

A hazai piacra nagyobb mennyiségben sokan hoznak be gépeket, de igazán jelentős szerepet vállaló a NOVOTRADE Rt. volt, mely az akkor újnak számító lízing segítségével vállalatok, szövetkezetek és sok egyéb felhasználó részére – igaz, a maihoz képest borsos áron – de megteremtette a széles körben alkalmazott számítástechnikát. Géppel, programmal, könyvekkel...

Ha eladták, hát jótállást kellett vállalni, sőt a 3 éves futamidejű lízing alatt még karbantartást, kiegészítő cikkek forgalmazását is stb... Ehhez viszont szerviz is kell.

Nos így jöttünk mi létre Fotoelektronik-Novotrade Gazdasági Társaság néven egy lelkes miskolci kolléga és néhány segítője után itt Budapesten, a hajdani alumíniumbolt helyén (még ma is keresik telefonon) itt, a Belvárosban a Magyar utca 12-ben.

A nagy áttörés 1985-től kezdődött. Rövid két év alatt igazán dicséretes országos szervizhálózat alakult ki, megfelelő lett az alkatrészellátás, létrejött egy szervizkultúra az ügyfelek fogadása, a gépek javítására, a munkalapok, raktárforgalom hatékony kezelésére és sorolhatnám tovább. Persze a C-16-os gépek, vagy a +4-esek inváziója idején tízméteres sorokban állók ebből csak a kényszerű bosszúságot látták. Aztán mi is átalakultunk és szorgos-serény NOVOTRADE Szerviz Kft. lettünk.

Magánügyfeleink csak rendre hozták az alapgépet, magnót, floppyt és a tápegységet, meg a joystickot. Gyerek magyarázta, hogy a „program nem jön be", szülő mérgelődött „micsoda dolog ez, hogy már megint mennyit kell fizetni", de mégis kialakultak azok a kedves kapcsolatok, ahol a telefonban nem a „szerelőt" kérték, hanem a Robit, Gézát, Ákost, Zsoltit... vagy „azt a barna hölgyet aki a gépet veszi be és nekem megígérte, hogy mára kész lesz"...

A vállalatok, intézmények részére kedvezően alakult az átalánydíjas javítási és karbantartási szerződésforma, ami folyamatos üzembiztonságot adott és az előírt javítási határidőre, vagy attól eltérő esetekben cseregép biztosításával, de mindig a partner érdekeit szem előtt tartva biztosította a gépek működőképességét. Volt is olyan év – nem is egy –, amikor több mint tízezer munkalapot kezeltünk és egyes esetekben ez több egységet is jelenthetett. A kiszedett IC-k tucatnyi kilóit adtuk át nemesfém-visszanyerésre és javában folyt a nyomtatófejek és motorok felújítása, meg a tápegységek gyártása.

Aztán egyszer csak valami megváltozott.

A rendszer, a gazdaság, a környezet a pénztárca és a hangnem is. Kevesebb lett a garanciális kötelezettség: a közületi partnerek is átalakultak, megszűntek, gépeiket lecserélték, eladták.

Mi is váltottunk, de a piac gyorsabb volt. Magánügyfeleink sem igen tudták elfogadni a 25% ÁFA-t, és az alkatrészbeszerzésre fordított márka 20-ról 50...55-re felkúszó „piaci" korrekcióját, a VÁM-ot, statisztikai illetéket és egyebeket.

Régen nem jut már szerviz felújításra, halványabban pislognak az éppen leszakadni akaró neonfények is a mennyezeten és virág helyett szemét és kutyapiszok kerül reggelente az utcai virágtartókba.

Tevékenykedünk, javítunk, adunk és veszünk, de valahogy már nem ragyognak úgy a régről ismert és titkon újra várt fények.

Bezárt a pécsi szerviz, majd megváltunk Salgótarjántól, Győrtől, Kaposvártól, a kecskeméti felvevőtől, a kispesti szerviztől és magányos, egyszemélyes lett Székesfehérvár meg Debrecen is.

Az idei karácsony egy lakatot tesz a Magyar utcai ajtóra is. Talán már ma is veszteséges, vagy csak holnap lesz az?

Keressünk-e új utakat, vagy embereket, hitet és erőt vagy új mosolyt az arcokra? Már nem... a törülközőt bedobták és a sokezernyi ügyfél ezentúl a Józsefváros mélyén, a panelházak szélén kap gyógyírt ezernyi bajára, kedvenc Commodore-jára.

Így búcsúzik az elmúlt évek valamennyi itt dolgozójának nevében az utolsó szervizvezető:

Kertész István

Olcsó C-64-es programok eladása kazettán. Játék programok, felhasználói programok, oktatói programok. Salamon Péter, 4122 Gáborján, Fő u. 21.

Commodore, PC, Epson,, Star, TVC gépek javítása, Epromok égetése. Cím: Budapest XVI., Ködös u. 42. Tel.: 2714-186.

Eladó alig használt Commodore 128-as számítógép + 1571-es lemezmeghajtó + lemezek + szakkönyvek. Irányár 28 000 Ft. Érdeklődni lehet az 1-353-900-as telefonszámon Szőke Miklósnénál.

C-64-re új játék- és felhasználói programok eladók. Bélyeggel ellátott válaszborítékért listát küldünk. (Vízhatlan csomagolás, kedvezmények!) FRIEND TWO CREW, 1399 Bp., Pf.: 701/55. Ezúton köszönetet mondunk a HERTA Kereskedelmi Kft.-nek, valamint a FŐVÁROSI SZÁLLÍTÁSI TANÁCSADÓ BT.-nek.

C-64-re cserélek programokat, csak lemezen. Listát kérek és küldök. Czifra Zoltán, 5231 Fegyvernek, Kiss János út 18/a.

Keresem Amigára a MIG-29-M Super FULCAUM nevű játékot. Hajdany Ádám, 1145 Budapest, Újvilág u. 39./b.

C-16 és Plus/4 szuperprogramok olcsón eladók kazettán és lemezen. Kérj listát! Bende Ferenc, 8500 Pápa, Attila u. 5/b.

C-64 II + 1541 II + FINAL III + magnó + 12 kazetta + 50 lemez + lemeztartó. Sürgős! A500 plusra cserélhető! Érdekel használt A500 plus is. Sürgős!! Berecz Tamás, tel.: 06-36/356-459.

C-64-re járékprogramok eladók. A játékprogramok darabja 15 Ft. Válaszboríték ellenében listát küldök. Sipos Csaba, 3770 Sajószentpéter, Sport út 7. 3/2.

C-64 DISK-re PROGRAMHEGYEK eladók! Régi és új játék és felhasználói programok, szakleírások, GEOS-ok eredeti állapotban, OLCSÓ ÁRON! Tájékoztatóhoz és listához szükséges 1 db lemez és 20 Ft-os bélyeg, vagy lemez nélkül 70 Ft. Holecz Ildikó, 1045 Budapest, Rózsa u. 21. VI/24.

C-64-re már 3 Ft-os ártól eladók kiváló játék, felhasználói és egyéb programok, lemezre és kazettára egyaránt, leírásokkal. Bélyegzett válaszborítékért listát küldök. Kovács Krisztián, 6750 Szeged-Algyő, Bartók Béla u. 6.

C-64-es programok eladók lemezen. Csere is érdekel. Érd: (Pád Softwer) 6772 Deszk, Alkotmány u. 4. Tel.: (62) 372-491.

C-64-re programok eladók lemezre és kazettára, kedvezménnyel. Felbélyegzett válaszborítékért listát küldök. Papp Mihály, 4034 Debrecen, Sterbinszky u.37.

A játékhoz ÖRÖKÉLET kell. POKE LEXIKON a CODI különkiadásában. Megrendelhető: Varsányi, 9700 Szombathely, Nagy László u. 11.

Amit tudni kell a GEOS-ról. Az első GEOS-LEXIKON. Ára: 200.- Ft + postaköltség. Megrendelhető: Varsányi, 9700 Szombathely, Nagy László u. 11.

Még kapható a CODI évkönyv (3 lemezen) 255 Ft utánvéttel. Varsányi, 9700 Szombathely, Nagy László u. 11.

GIGA-CAD leírás lemezen 150 Ft + nyomtatva 450 Ft + postaköltség. Cím: Apró Lajos, 4220 Hajdúböszörmény, Táncsics u. 72.

C-64-re sikerprogramok reklámáron, kazettán és lemezen eladók. Írjatok gyorsan! Bende Ferenc, 8500 Pápa, Attila u. 5/b.

C-64-ed van? BASIC-et tanulsz? Akkor itt az alkalom: BASIC OKTATÓ 1.: alapbasic oktató + 10 példaprogram + adathordozó: 250 Ft + portó. BASIC OKTATÓ 2.: alapbasic oktató + 15 példaprogram + az összes BASIC utasítást tartalmazó, nyomtató BASIC LEÍRÁS + magyar nyelvű hibaüzenetek program + adathordozó: 300 Ft + portó. Az oktató és a leírás ékezetes. Válaszborítékért információ. Megrendelhető utánvéttel. Szűcs Gábor, 3528 Miskolc, Jendrassik u. 10. Tel.: (46) 389-018.

Eladó 120 MB játék és felhasználói program PC-re 3M 5'25"-ös és 3'5"-es lemezeken. Válaszborítékra tájékoztató. Nagy István, 1202 Budapest, Mártírok útja 151.

PC AT programokat cserélek (főleg SVGA). Küldj 5'25"-ös lemezeket, amelyeket a saját programjaimmal garantáltan visszaküldök! Vannak szuper zeneprogramok, és csúcsszuper grafikai demofelvételek, de a játékaim sem elhanyagolhatók. Ezen kívül rendelkezem a legjobb vírusölőkkel és immunizáló programokkal is. Megvannak a legelterjedtebb csehszlovák felhasználói és szövegszerkesztő programok. Választhatsz az év legjobb programjaiból. Külön kérésre listát is küldök. Ha kedvet kaptál, de ha nem, akkor is. Küldj programokat a következő címre: Vigh Attila, Jantárová 1868/13 92901 Dunayská Streda Szlovákia. Jelige: csak vírusmentes programokat várok és küldök.

Hi Amigások! Akartok 1250 db lemezt szinte ingyen. Ha igen akkor válaszboríték ellenében elküldöm, hogy mit kell tenned a lemezekért. Cím: Coddy, 8000 Székesfehérvár, Benke F. u. 13. fszt. 1. Természetesen C-64-re is fennáll a lehetőség, hogy megszerezd az 1250 db lemezt, teli szuper programokkal. Ha érdekel, akkor tüntesd fel a géped típusát, és küld el a felső címre. Még valami, ha akarsz C-64-re 64 Kbytes, vagy 32 Kbytes kartridge-t zenélős főmenűvel a lehető legolcsóbb áron, akkor is írj T.K.S., a cím ugyan az mint fenn. Még mindig, ha akarsz 4 db 8 Kbytes epromot szintén írj. CODDY OF TKS. Bye!

C-64-es programok eladók lemezen és eladó egy C-64-hez való magnó. Érdeklődni: Pád Softwer, 6772 Deszk, Alkotmány u. 4.

C-64-re 93-ban is A.Comp.W-től rendelje a legújabb, legjobb programokat. Új DS/DD lemezeken csak 80 Ft/db! Listát küldünk! A.Comp.W, 3014 Hort, Pf.: 2.

C-128/64 gépre programokat adok — veszek — cserélek. 10.000 programom van. Cím: Járóka László, 1148 Budapest, Adria sétány 6. L/I. 2.

C-64-esek figyelem! Lemezeket és programokat adok el olcsón! A lemezeknek 40 Ft darabja, de 20 db már csak 7503 Ft. Játékokat is adok el amelyekből itt egy pld.: Elvira 2, Creatures 1., 2., stb.! Válaszborítékért listát és tájékoztatót küldök. A fizetés módja postai utalványon történik! Egyébként GEOS-t keresek! Címem: Oláh Attila, 5900 Orosháza, Szarvasi út 62.

Vennék C-16 alapgépre játék, oktató és szövegszerkesztő programot kazetta cartridge. Lőrik Ernő, 3524 Miskolc, Hajós u. 4.

ACOMP Kft.

FEBRUÁRI
60 Ft-os
vásárlási utalványa

Beváltható
készpénzes vásárlás esetén a
1141 Budapest, Álmos vezér útja 17.
szám alatti üzletben.

Érvényes: 1993. február 28-ig.

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

A NOVOTRADE SZERVÍZ Kft. az alább felsorolt szervízeiben mindenféle szervízszolgáltatás munkadíjából 10% kedvezményt ad az egyesületi tagoknak.

1083 Budapest, Szigony u. 9.	Tel.: 134-3153
3525 Miskolc, Fazekas u. 1—3.	Tel.: 46/321-488
5600 Békéscsaba, Bartók B. u. 37.	Tel.: 66/327-195
6724 Szeged, Csongrádi sugárút 76.	Tel.: 62/313—377
8000 Székesfehérvár, Rádió u. 15/A.	Tel.: 22/319-765
9700 Szombathely, Szalonak u. 31.	Tel.: 94/14-519

NOVOTRADE SZERVÍZ Kft.

Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal.
A kedvezmény többször is igénybe vehető.

MAKROVILÁG
utazási iroda

Beváltható utazás megrendelése esetén

az Üllői úti főirodában az alábbiak szerint:

5 000 Ft-ig	— 200 Ft kedvezmény
10 000 Ft-ig	— 400 Ft kedvezmény
20 000 Ft-ig	— 500 Ft kedvezmény
20 000 Ft felett	— 1000 Ft kedvezmény

Csoportok jelentkezése esetén további kedvezményekről az irodában lehet tárgyalni

NOVOTRADE

FEBRUÁRI
60 Ft-os
vásárlási utalvány

ÉRVÉNYTELEN

Beváltható készpénzes vásárlás esetén a 2C Áruházban.
Bp. XIII., Balzac u. 35.

Érvényes: 1993. február 28-ig.

HOBBI ELEKTRONIKA

FEBRUÁRI
vásárlási utalványa

Értéke:

5000 Ft-ig 80Ft,
5000 Ft felett 10%

Beváltható a Hobbi Elektronika Kft.-nél.
Budapest VII., Dózsa György u. 16.
Telefon: 122-8892
Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

Az Országos Commodore Egyesület szolgáltatásai

Egyesületi tagoknak 20% kedvezmény:

VC—20 memóriabővítés 3—27 kByte-os:	kiépítéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	5000 Ft
FÉK C—16, C—116, +4 potméteres sebességváltoztatás 0%-tól 100%-ig fokozatmentesen	2000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	4300 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépítésekben:	
— 8 kByte SOFT—ROM	3400 Ft
— 16 kByte SOFT—ROM	4000 Ft
— 8 kByte SOFT—ROM 16 kByte EPROM	4400 Ft
— 16 kByte SOFT—ROM 16 kByte EPROM	5000 Ft
— 16 kByte EPROM	2200 Ft

Egyesület tagoknak 30% kedvezmény:

Speeddos (átkapcsolható) operációs rendszer beépítése (C64 átalakítás, lemezegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése	900 Ft
PAGEFOX magyar ékezetes kiadvány-szerkesztő cartridge (a teljes A/4-es oldal kinyomtatásához 640 pont/soros nyomtató szükséges minimum, pl. Citizen 120D)	5500 Ft
FASTLOAD cartridge (lemezes gyorstöltő, másoló, monitor)	1500 Ft
TTL IC-teszter cartridge + program	4300 FT
288/256 Kbyte-os eprombank (vezérlő epprommal)	5000 Ft
Eprométegető (2716-tól 27512-ig)	5000 Ft
C64-hez tároló oszcilloszkóp	8000 Ft
C64-bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256 K RAM-diszk (256 Kbyte RAM-mal)	14000 Ft
256 K RAM-diszk (64 Kbyte RAM-mal)	9000 Ft
2×64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4.300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységesen	700 Ft

Árainkat az alkatrészárak változásai befolyásolhatják.
A fenti bővítések megrendelhetők levélben az O.C.E. címén, valamint személyesen a havonta rendezendő klubdélelőttön, ahol rendszeres bemutatót is tartunk.

Postázás esetén 100 Ft postaköltség kerül felszámításra.

A kedvezmény igénybevételéhez az O.C.E. tagsági igazolvány bemutatása szükséges.